



Miguel José Parreira Alves

Licenciado em Ciências de Engenharia Eletrotécnica e Computadores

Localização e Dimensionamento Ótimo de Bancos de Condensadores em Redes de Energia Elétrica

Dissertação para obtenção do Grau de Mestre em Engenharia Eletrotécnica e
Computadores

Orientador: Professor Doutor Francisco Alexandre Ganho da Silva Reis,
Professor Auxiliar Convidado, Faculdade de Ciência e Tecnologia, da
Universidade Nova de Lisboa

Co-Orientador: Professor Doutor Mário Fernando da Silva Ventim Neves,
Professor Auxiliar, Faculdade de Ciência e Tecnologia, da Universidade Nova de
Lisboa

Júri:

Presidente: Prof. Doutor Luís Filipe Figueira de Brito Palma

Arguente: Prof. Doutor Pedro Miguel Ribeiro Pereira

Vogal: Prof. Doutor Francisco Alexandre Ganho da Silva Reis



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2015

Localização e Dimensionamento Ótimo de Bancos de Condensadores em Redes de Energia Elétrica

**Copyright © Miguel José Parreira Alves, Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa**

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimento

Gostaria de deixar umas palavras de agradecimento aos que me apoiaram e contribuíram para a realização desta dissertação, bem como todo o meu percurso académico.

Agradeço à minha família, em particular aos meus pais, por todo o apoio e incentivo que recebi ao longo da minha vida académica.

Agradeço a todos os professores que contribuíram para a minha formação académica, em especial, ao Professor Doutor Francisco Reis, pela sua orientação, disponibilidade e exigência no decorrer deste trabalho.

Para os amigos e namorada, que me apoiaram e motivaram a completar esta fase da minha vida, um muito obrigado.

Por fim, agradeço a todos os colegas que conheci, com quem tive oportunidade de conviver e trabalhar ao longo de todo o curso.

Resumo

O problema endereçado nesta dissertação é o desenvolvimento de métodos de otimização que contribuam para a redução de perdas elétricas, nas redes elétricas, mediante a seleção e a correta implementação de bancos de condensadores. Assim, o objetivo desta dissertação é desenvolver uma metodologia para melhorar a eficiência energética das redes elétricas por meio da redução das perdas elétricas. O problema consiste em encontrar a capacidade e localização ótima na rede para a implementação de bancos de condensadores.

Visto este problema ser de natureza combinatória, foi adotado o algoritmo de otimização *PSO* (*Particle Swarm Optimization*) e o algoritmo *EPSO* (*Evolutionary Particle Swarm Optimization*) para o resolver. O algoritmo *EPSO* é uma evolução do algoritmo *PSO*. O programa foi desenvolvido na linguagem de programação *Python*, no ambiente *Eclipse*. Cada partícula, que corresponde a uma solução do problema, é avaliada com recurso ao trânsito de energia resolvido no ambiente de simulação disponibilizado pelo programa *PSS/E*.

As redes elétricas, que serviram como casos de estudo, foram as redes “IEEE6”, “IEEE14” e outra rede de vinte e um barramentos, rede que representa uma topologia típica de produção, transporte e distribuição de energia elétrica.

Concluiu-se que ambos os algoritmos cumpriram com o objetivo do trabalho planeado. Para todas as redes, os algoritmos conseguiram encontrar soluções que reduzissem as perdas elétricas totais da rede elétrica. De facto, o *EPSO* revelou-se mais eficaz na pesquisa de soluções mais adaptadas ao problema. Para garantir a convergência para o ótimo local do espaço de pesquisa, os parâmetros que se revelaram mais significativos foram o número de partículas da população e o número de gerações.

Palavras chave: perdas elétricas, banco de condensadores, *particle swarm optimization*, *evolutionary particle swarm optimization*, trânsito de energia.

Abstract

The matter dealt in this thesis is to develop optimized methods which may contribute to the reduction of electric energy losses in electrical networks by choosing and implementing correctly capacitor banks. Therefore, the objective of this work is to develop a methodology in order to improve the energy efficiency of electrical networks by reducing the electrical losses. The problem is to find the optimal sizing and siting for the implementation of capacitor banks.

As this problem is of a combinatory nature, the algorithms *PSO* (Particle Swarm Optimization) and *EPSO* (Evolutionary Particle Swarm Optimization), which is an evolution of the PSO, were adopted to solve it. The program was developed in *Python* programming language together with the *Eclipse* programming platform. Each particle corresponding to a solution of the problem is evaluated using the power flow solved in the simulation framework provided by the *PSS/E* program.

The electrical networks “IEEE6”, “IEEE14” were the ones used as case studies as well as a network of twenty-one buses. This one represents a typical network topology comprising generation, transmission and distribution of electricity.

It was concluded that both algorithms fulfilled the purpose of the planned work. For all network, the algorithms were able to find out solutions that reduce the total electrical losses. The EPSO algorithm proved to be more effective in finding better adapted solutions to the problem. To ensure convergence to the optimal location of the search space, the parameters that have proved to be most significant were the number of particles of the population and the number of generations.

Keywords: electrical losses, capacitor bank, particle swarm optimization, evolutionary particle optimization, power flow.

Índice

1. Introdução	1
1.1 Enquadramento	2
1.2 Objetivos	3
1.3 Contribuições	3
1.4 Estrutura da tese	4
2. O problema da Localização e Dimensionamento de Bancos de Condensadores	5
2.1 Estado de arte	6
2.1.1 Métodos de redução de perdas de energia elétrica	7
2.1.2 Natureza do problema	9
2.1.3 Técnicas e algoritmos de otimização para o problema	10
2.2 Formulação do problema	12
2.3 Modelização	14
2.3.1 Trânsito de energia	14
2.3.2 Cálculo das equações do trânsito de energia	17
3. Metodologias de otimização	21
3.1 Meta-heurística	22
3.2 Computação Evolucionária	23
3.2.1 Estratégias de Evolução e Programação Evolucionária	24
3.2.2 Algoritmos Genéticos	28
3.3 Otimização por Enxame de Partículas	31
3.3.1 Objetivo da <i>PSO</i>	31
3.3.2 Descrição do algoritmo	31
3.4 Otimização por Enxame de Partículas Evolucionário (<i>EPSO</i>)	34
3.4.1 Objetivo da <i>EPSO</i> e suas origens	34
3.4.2 Descrição do algoritmo	35
3.5 Partícula	36
3.5.1 Definição de partícula	36

3.5.2	Variáveis de decisão e parâmetros estratégicos da partícula	38
3.5.3	Avaliação da partícula	40
4.	Aplicação dos algoritmos	41
4.1	Arquitetura do programa.....	42
4.2	Algoritmos aplicados	44
4.2.1	Aplicação do algoritmo <i>PSO</i> ao problema <i>OCP</i>	44
4.2.2	Aplicação do algoritmo <i>EPSO</i> ao problema <i>OCP</i>	45
4.3	Parametrização dos algoritmos	46
4.3.1	<i>PSO</i>	48
4.3.2	<i>EPSO</i>	52
4.4	Aplicação dos algoritmos às redes em estudo	54
4.4.1	Rede “IEEE6”	54
4.4.2	Rede “IEEE14”	58
4.4.3	Rede de vinte e um barramentos	66
4.5	Eficiência e Robustez dos algoritmos	74
4.5.1	Rede de catorze barramentos	74
4.5.2	Rede de vinte e um barramentos	76
5.	Conclusão	79
5.1	Observações finais	80
5.2	Perspetivas para trabalhos futuros.....	81

Índice de Figuras

Figura 1.1: Perdas nas redes de transporte e distribuição em Portugal.	2
Figura 2.1: Gráfico da compensação de energia reativa	8
Figura 2.2: Condensador ideal para redução de perdas usando a regra dos dois terços.	10
Figura 2.3: Diagrama de linha única de uma fonte principal.	13
Figura 2.4: Modelização das linhas e barramentos de uma rede elétrica.	14
Figura 2.5: Potência transitada numa linha	16
Figura 2.6: Esquema equivalente em π de uma linha de transmissão.	18
Figura 2.7: Fluxograma do método Newton-Raphson [17].	19
Figura 3.1: O ciclo básico da evolução [22].	24
Figura 3.2: Representação de um indivíduo i com n variáveis reais.	25
Figura 3.3: Algoritmo Estratégias de Evolução [23].	26
Figura 3.4: Algoritmo geral da Programação Evolucionária [23].	27
Figura 3.5: Aplicação de um AG num problema de navegação robótica [28].	28
Figura 3.6: Fluxograma que exemplifica o funcionamento do Algoritmo Genético [28].	30
Figura 3.7: Ilustração das três componentes do movimento de uma partícula i .	32
Figura 3.8: Topologias de Enxame. a) Melhor global. b) Topologia em anel. c) Topologia radial.	33
Figura 3.9: Fluxograma que exemplifica o funcionamento da PSO.	34
Figura 3.10: Ilustração das três componentes do movimento de uma partícula i . Relativamente ao termo de cooperação, a atração efectua-se para uma vizinhança.	36
Figura 3.11: Caracterização simples de uma partícula.	37
Figura 3.12: Caracterização detalhada de uma partícula.	37
Figura 3.13: Gráficos de convergência das partículas.	38
Figura 4.1: Arquitetura do programa englobando a arquitetura da aplicação.	43
Figura 4.2: Fluxogramas dos procedimentos principais das duas abordagens da PSO.	44
Figura 4.3: Fluxogramas dos procedimentos principais dos dois métodos da EPSO.	45
Figura 4.4: Esquema da rede original de seis barramentos.	47
Figura 4.5: Gráfico de convergência de um teste da segunda fase de testes.	50
Figura 4.6: Gráficos de convergência para diferentes velocidades.	51
Figura 4.7: Diferença entre o peso do parametro estratégico inércia do algoritmo PSO e EPSO com o avançar das gerações.	54
Figura 4.8: Esquema da rede original de seis barramentos.	55
Figura 4.9: Esquema da rede de seis barramentos, depois de aplicado o algoritmo PSO.	56
Figura 4.10: Perfil de tensões da rede original de seis barramentos e da mesma rede com implementação de condensadores.	58
Figura 4.11: Esquema da rede original de catorze.	59
Figura 4.12: Esquema da rede de catorze barramentos depois de aplicado o algoritmo PSO.	61
Figura 4.13: Esquema da rede de catorze barramentos depois de aplicado o algoritmo EPSO.	63

<i>Figura 4.14: Perfil de tensões da rede original de catorze barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo PSO.</i>	65
<i>Figura 4.15: Perfil de tensões da rede original de catorze barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo EPSO.</i>	65
<i>Figura 4.16: Esquema da rede original de vinte e um barramentos.</i>	67
<i>Figura 4.17: Esquema da rede de vinte e um barramentos depois de aplicado o algoritmo PSO.</i>	69
<i>Figura 4.18: Esquema da rede de vinte e um barramentos depois de aplicado o algoritmo EPSO.</i>	71
<i>Figura 4.19: Perfil de tensões da rede original de vinte e um barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo PSO.</i>	73
<i>Figura 4.20: Perfil de tensões da rede original de vinte e um barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo EPSO.</i>	73
<i>Figura 4.21: Eficiência e robustez do algoritmo PSO relativamente à rede de catorze barramentos.</i>	75
<i>Figura 4.22: Eficiência e robustez do algoritmo EPSO relativamente à rede de catorze barramentos.</i>	75
<i>Figura 4.23: Eficiência e robustez do algoritmo PSO relativamente à rede de vinte e um barramentos.</i>	76
<i>Figura 4.24: Eficiência e robustez do algoritmo EPSO relativamente à rede de vinte e um barramentos.</i>	76
<i>Figura 4.25: Diferença entre o peso do parâmetro estratégico inércia do algoritmo PSO e EPSO com o avançar das gerações.</i>	77
<i>Figura 4.26: O peso dos parâmetros estratégicos memória e cooperação do algoritmo EPSO com o avançar das gerações.</i>	78

Índice de Tabelas

<i>Tabela 1: Perdas na rede versus procura total de eletricidade [3].....</i>	<i>6</i>
<i>Tabela 2: Tipos de barramentos.</i>	<i>17</i>
<i>Tabela 3: Parâmetros escolhidos para a primeira fase de testes.</i>	<i>48</i>
<i>Tabela 4: Resultados obtidos na primeira fase de testes utilizando o algoritmo PSO.</i>	<i>48</i>
<i>Tabela 5: Parâmetros escolhidos para a segunda fase de testes.</i>	<i>49</i>
<i>Tabela 6: Resultados obtidos na segunda fase de testes utilizando o algoritmo PSO.</i>	<i>49</i>
<i>Tabela 7: Parâmetros escolhidos para a terceira fase de testes.</i>	<i>50</i>
<i>Tabela 8: Resultados obtidos na terceira fase de testes utilizando o algoritmo PSO.</i>	<i>51</i>
<i>Tabela 9: Resultados obtidos na primeira fase utilizando o algoritmo EPSO.</i>	<i>52</i>
<i>Tabela 10: Parâmetros escolhidos para a segunda fase utilizando o algoritmo EPSO.</i>	<i>52</i>
<i>Tabela 11: Resultados obtidos utilizando o algoritmo EPSO.</i>	<i>53</i>
<i>Tabela 12: Melhor solução encontrada pelo algoritmo PSO, referente à rede de seis barramentos.</i>	<i>55</i>
<i>Tabela 13: Parâmetros escolhidos para o algoritmo PSO quando aplicado à rede de catorze barramentos. ..</i>	<i>60</i>
<i>Tabela 14: Resultados obtidos das aplicações do algoritmo PSO à rede de catorze barramentos.</i>	<i>60</i>
<i>Tabela 15: Resultados obtidos das aplicações do algoritmo EPSO à rede de catorze barramentos.</i>	<i>62</i>
<i>Tabela 16: Parâmetros escolhidos para o algoritmo PSO quando aplicado à rede de vinte e um barramentos.</i>	<i>68</i>
<i>Tabela 17: Resultados obtidos das aplicações do algoritmo PSO à rede de vinte e um barramentos.</i>	<i>68</i>
<i>Tabela 18: Resultados obtidos das aplicações do algoritmo EPSO à rede de vinte e um barramentos.</i>	<i>70</i>

Acrónimos

AA – Auto Adaptativos

AG – Algoritmos Genéticos

API – Application Program Interface

AT – Alta Tensão

EE – Estratégias de Evolução

EPSO – Evolutionary Particle Swarm Optimization

FCT – Faculdade de Ciências e Tecnologia

IEEE – Institute of Electrical and Electronics Engineers

INESC – Instituto de Engenharia de Sistemas e Computadores

MAT – Muito Alta Tensão

OCP – Optimal Capacitor Placement

PE – Programação Evolucionária

PSO – Particle Swarm Optimization

PSS/E – Power System Simulator for Engineering

SVC – Static VAR Controller

UNL – Universidade Nova de Lisboa

1. Introdução

Exposição do tema da dissertação
e apresentação das soluções
para o problema que é endereçado.

1.1 Enquadramento

Associado à evolução da tecnologia no mundo atual, cresce o aumento do consumo de energia elétrica, tornando a eletricidade um bem essencial para a sociedade. Se, numa fase inicial, as principais redes elétricas de energia eram caracterizadas por terem pequenas dimensões e lidarem com reduzidas potências, esta crescente procura de energia elétrica assim como a necessidade de recorrer a energias renováveis, muitas vezes localizadas em zonas afastadas dos locais de consumo, torna o problema de desenvolvimento de redes cada vez mais complexo.

Para garantir a fiabilidade e flexibilidade do sistema elétrico de energia bem como as condições necessárias de segurança com o intuito principal de fornecer um serviço de qualidade aos consumidores, certos desafios são propostos ao sistema elétrico, a saber:

- Aumentar a eficiência energética na operação das redes elétricas;
- Melhorar a qualidade de serviço oferecido e a segurança de fornecimento;
- Reduzir as emissões poluentes, resultantes da produção de eletricidade;
- Aumentar a produção de energia renovável e produção distribuída de energia elétrica.

A diminuição das perdas elétricas numa rede elétrica é a base para o aumento da eficiência energética nas redes. Vários estudos têm mostrado que a redução das perdas nas redes elétricas tem-se revelado um método vantajoso para elevar a eficiência energética nessas redes. É nos sistemas de distribuição da rede elétrica que ocorrem as maiores perdas elétricas, como se pode constar pela Figura 1.1 [1], [2].

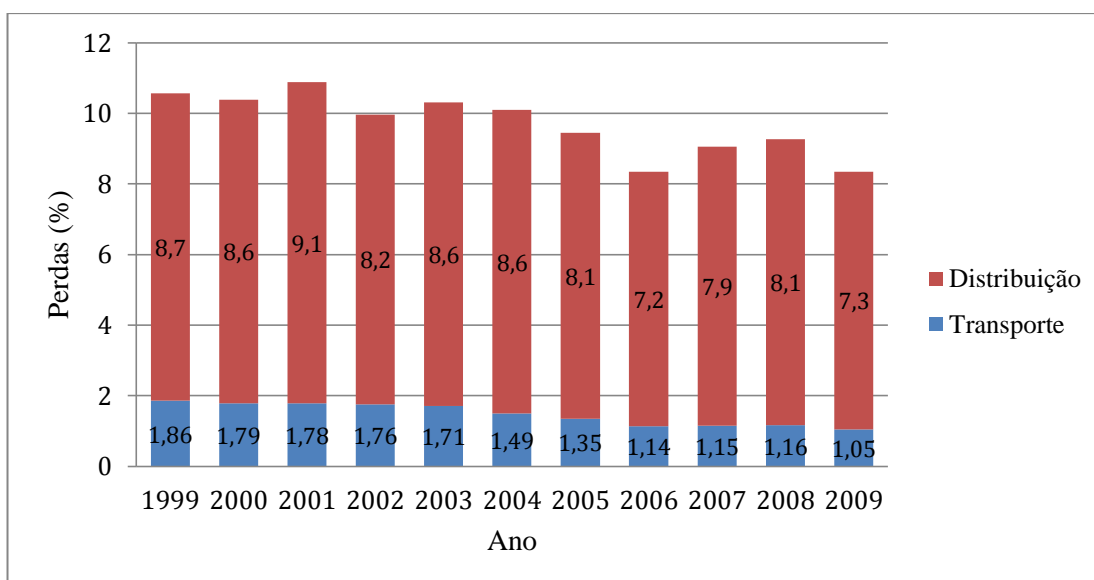


Figura 1.1: Perdas nas redes de transporte e distribuição em Portugal.

Apesar de a redução das perdas elétricas na rede elétrica estar principalmente ligada à eficiência energética, este recurso também apresenta uma grande importância do ponto de vista ambiental, ou seja, a redução das perdas elétricas significa uma redução de produção de eletricidade nas centrais, pois não será necessário produzir essa energia que serviria para alimentar essas perdas, diminuindo assim as emissões poluentes.

Existem vários métodos de redução das perdas elétricas, sendo eles a compensação de potência reativa, o reforço das linhas, a mudança do nível de tensão, a reconfiguração da rede, a produção distribuída, entre outros.

1.2 Objetivos

Como foi referido na secção 1.1, a redução das perdas elétricas na rede elétrica desempenha um papel importante na otimização da rede elétrica e pode ser efetuada de diferentes formas. Neste trabalho, o método adotado foi a compensação de potência reativa através da implementação de bancos de condensadores. O problema a otimizar consiste em encontrar a melhor localização (barramentos da rede) e capacidade dos bancos de condensadores que se devem implementar na rede a fim de reduzir as perdas elétricas.

Nesse sentido, foi adotado o algoritmo *PSO* (*Particle Swarm Optimization*) e uma evolução desse mesmo algoritmo, o *EPSO* (*Evolutionary Particle Swarm Optimization*), para resolver o problema *OCP* (*Optimal Capacitor Placement*). Estes algoritmos foram desenvolvidos de modo a estarem aptos para serem aplicados a qualquer rede de qualquer topologia. Pretende-se identificar, igualmente, quais os fatores que influenciam o desempenho dos algoritmos de forma a obter uma correta parametrização dos mesmos. O desenvolvimento dos algoritmos recorre, por um lado, ao *software* de simulação de redes, *PSS/E*, para avaliação de cada partícula e por outro à linguagem *Python* para a implementação dos algoritmos.

1.3 Contribuições

As previsões apontam que o consumo de energia elétrica irá aumentar no futuro e como consequência o impacto ambiental irá ser maior. Assim, a possibilidade de realizar este estudo (desenvolvimento dum programa capaz de minimizar as perdas totais de uma rede elétrica) poderá contribuir para promover a eficiência energética nas redes elétricas.

No decorrer dos tempos, vários algoritmos de otimização foram utilizados para resolver este problema. Com este estudo pretende-se analisar e comparar os resultados provenientes da aplicação dos algoritmos *PSO* e *EPSO* a este problema de otimização.

Através da aplicação destes algoritmos, será possível saber os barramentos onde a implementação de bancos de condensadores irá ser mais determinante para reduzir as perdas elétricas totais da rede.

1.4 Estrutura da tese

A presente dissertação é constituída por cinco capítulos que, por sua vez, se dividem em vários subcapítulos.

O primeiro capítulo é composto por uma parte introdutória, onde é feito o enquadramento do problema e apontados os principais objetivos deste trabalho.

O segundo capítulo consiste na apresentação do problema, onde é identificada a natureza do problema e é feita a formulação do mesmo. São apresentadas potenciais formas de resolução para este tipo de problemas.

No terceiro capítulo é apresentada a metodologia para resolução do problema. É feita uma introdução teórica acerca das bases dos algoritmos aplicados, bem como a apresentação pormenorizada dos mesmos. O conceito partícula é explicado, referenciando a sua grande importância no desempenho dos algoritmos aplicados. Por fim, é feita a modelização da rede elétrica e apresentado o método iterativo usado para a resolução das equações do trânsito de energia.

O quarto capítulo reside na aplicação dos algoritmos a redes elétricas escolhidas. As redes elétricas escolhidas, que serviram como casos de estudo, foram redes de seis, catorze e vinte e um barramentos. É feita uma análise aos resultados obtidos depois de aplicados os algoritmos face aos resultados da rede elétrica original.

O quinto e último capítulo destina-se às conclusões acerca do trabalho efetuado. São igualmente identificadas propostas para trabalhos futuros relacionados com o tema abordado nesta dissertação.

2. O problema da Localização e Dimensionamento de Bancos de Condensadores

Formulação do problema e
apresentação das metodologias
capazes de resolver o problema *OCP*.

2.1 Estado de arte

A análise de distribuição de energia elétrica é uma importante área de pesquisa devido ao facto desta ser a fase final na entrega de eletricidade aos consumidores a partir de sistemas de transmissão em muito alta tensão (*MAT*). Os sistemas de distribuição de energia elétrica estão cada vez maiores e mais complexos, fazendo com que os fluxos de potência reativa aumentem causando perdas elevadas. Anualmente, as perdas de energia elétrica na transmissão e distribuição em média rondam os 7% na União Europeia [3].

Na tabela 1 é apresentada a evolução das perdas na rede elétrica na União Europeia. Assumindo que 2% é de perdas na transmissão e 6 a 8% é na distribuição, isto representa 7 mil milhões de Euros em energia desperdiçada, cada ano, na distribuição. Estes números incluem as perdas nas linhas em média e baixa tensão, como também nas subestações primárias e secundárias. Quando as potências das cargas, numa determinada rede, são elevadas, as perdas podem até ser mais significativas, o que resulta numa elevada queda de tensão em certas regiões da rede de distribuição. O perfil de tensões reduzido nos barramentos é devido a grandes perdas de transmissão que são inerentes, devido à resistência das linhas de transmissão, e devido às correntes que fluem através delas.

Tabela 1: Perdas na rede versus procura total de eletricidade [3].

	1980	1990	2000	2007	2008	2010	2020
Consumo Final (TWh)	1,703.6	2,175.4	2,633.0	2,928.0	2,938.9	2,595.0	3,242.2
Perdas na rede (TWh)	136.9	167.2	211.8	201.3	201.4	181.9	222.1
Perdas na rede (%)	7.4	7.1	7.4	6.4	6.4	6.6	6.4
Procura total de eletricidade (TWh)	1,840.6	2,342.6	2,845.2	3,126.1	3,136.7	2,771.5	3,464.3

Para assegurar um funcionamento eficiente, seguro e fiável do sistema de energia elétrica, é fundamental que as tensões nos barramentos sejam mantidas dentro dos limites toleráveis, para que os equipamentos de energia elétrica possam funcionar com segurança a uma tensão muito próxima das suas tensões nominais. Sendo assim, é importante para as empresas de distribuição otimizar o seu funcionamento, diminuindo as perdas por efeito de joule P , 2.1, onde I é a corrente elétrica que percorre o condutor com determinada resistência R . Melhorar o perfil de tensões, nos diferentes barramentos da rede, é outro pormenor importante na otimização das redes elétricas.

$$P = I^2 \cdot R \quad 2.1$$

Para garantir perdas mínimas e níveis adequados de tensão em diferentes pontos da rede de distribuição, a utilização de alguns dispositivos, que realizam o controlo efetivo da tensão, o controlo da potência reativa e o controlo do fator de potência é indispensável. Os Bancos de condensadores são um dos equipamentos utilizados para cumprir estes objetivos. Para alcançar os benefícios da redução de perdas por efeito de joule, a correção do fator de potência e a melhoria do perfil de tensões, sob várias restrições operacionais (limites de tensão dos barramentos, sobrecompensação, entre outros), é necessário determinar a localização e a capacidade dos condensadores a serem instalados para diferentes níveis de carga. A implementação de bancos de condensadores (*OCP*) é o método mais rentável e viável para melhorar o perfil de tensões e minimizar as perdas no sistema de transmissão, bem como fornecer o suporte de potência reativa para a operação confiável do sistema de energia.

2.1.1 Métodos de redução de perdas de energia elétrica

Além dos principais métodos de redução de perdas elétricas: compensação de energia reativa, substituição dos condutores, aumento do nível de tensão e reconfiguração da rede, há que salientar, devido à evolução do funcionamento dos sistemas de energia elétrica, a produção dispersa começa a desempenhar um papel importante nas redes de distribuição, podendo ser bastante útil para reduzir as perdas elétricas na rede.

Como foi referido no parágrafo anterior, existem vários métodos de redução de perdas elétricas. Neste trabalho apenas será abordado o método de compensação de energia reativa, mais propriamente, a implementação de bancos de condensadores.

Compensação de energia reativa

Esta é a forma tradicionalmente utilizada para reduzir as perdas e diminuir as quedas de tensão ao longo das linhas que constituem a rede. A energia reativa (caráter indutivo) é essencial para a criação de campos eletromagnéticos dos componentes do sistema. Esta energia circula através das linhas entre a fonte de alimentação e as cargas, diminuindo a capacidade de transmissão das linhas elétricas. O trânsito de energia reativa nas redes provoca um acréscimo na magnitude da corrente, e, consequentemente, um aumento no valor das perdas elétricas, visto que estas dependem quadraticamente da corrente. Assim, conseguindo uma diminuição no trânsito de potência reativa, haverá uma diminuição da intensidade da corrente elétrica nas linhas, que irá significar uma redução nas perdas.

A implementação de bancos de condensadores é a forma menos dispendiosa de introduzir potência reativa na rede. Os benefícios económicos, que surgem através da introdução destes aparelhos nas redes, são os seguintes:

- Na capacidade de geração libertada;
- Na capacidade de transporte libertada nas linhas e nas subestações de transformação;

- No sistema de distribuição, mais precisamente:
 - Diminuição das perdas nos condutores;
 - Melhoria na regulação da tensão;
 - Libertação da ocupação nas linhas de distribuição;
 - Aumento de faturação devido à melhoria da tensão na rede.

É habitual, na exemplificação deste método, ilustrar o triângulo de potências. O gráfico da compensação de energia reativa pode ser visto na Figura 2.1.

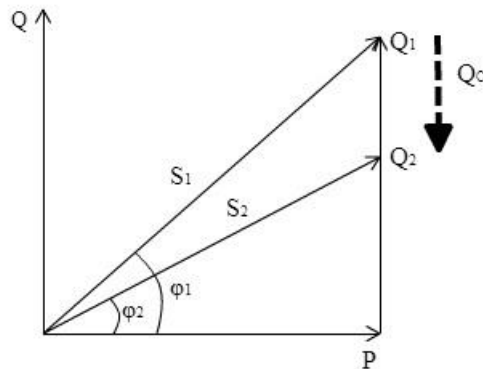


Figura 2.1: Gráfico da compensação de energia reativa

A redução de circulação de potência reativa na rede corresponde à potência de compensação, Q_C , que pode ser obtida pela seguinte expressão:

$$\vec{Q}_C = \vec{Q}_1 - \vec{Q}_2 = \vec{P}(\tan \varphi_1 - \tan \varphi_2) \quad 2.2$$

onde:

Q_C – potência reativa resultante da compensação (VAr);

Q_1 – potência reativa antes da compensação (VAr);

Q_2 – potência reativa depois da compensação (VAr);

φ_1 – ângulo de defasamento antes da compensação;

φ_2 – ângulo de defasamento depois da compensação;

P – potência ativa que circula na linha (W);

A energia reativa pode ser gerada em qualquer ponto da rede elétrica ao contrário da energia ativa, que tem de ser gerada nas centrais. Muitas vezes, na rede elétrica, existem vários bancos de condensadores instalados que deixam de exercer a sua função corretamente. Variações no perfil de carga, incertezas sobre valores das cargas, ou até uma má configuração dos seus controlos, podem explicar o facto dos bancos de condensadores não estarem a desempenhar a sua função corretamente.

Para estes casos, é fundamental encontrar a configuração adequada para cada banco de condensadores para que seja feita uma redução máxima possível das perdas elétricas da rede. Assim, o problema, quando é utilizado bancos de condensadores para a redução das perdas, pode ser definido como a procura da capacidade dos bancos de condensadores e a localização do melhor local para as suas implementações que se traduzirá na maior redução das perdas na rede.

Outros dispositivos de controlo de tensão

Para além da utilização dos clássicos condensadores, surgem alternativas como os compensadores estáticos do tipo *SVC* (*Static VAR Controller*), que mais que a compensação de energia reativa também conseguem a compensação de flutuações e cavas de tensão entre outras perturbações na rede, pois apresentam um poder de reação muito rápido.

Ao nível do transporte e da distribuição, a regulação de tensão pode também ser feita por meio de transformadores com comutação de tomadas. Estes permitem compensar variações de tensão que, inevitavelmente, ocorrem ao longo do sistema, através da alteração da relação de transformação aumentando ou diminuindo a tensão aos seus terminais. A variação da relação de transformação, que o comutador de tomadas proporciona, altera o fluxo de potência reativa entre os subsistemas ligados ao transformador, o que acarreta uma modificação do perfil de tensão.

2.1.2 Natureza do problema

O problema *OCP* suscita duas grandes questões que são a base do problema de otimização. A primeira é a localização ideal dos bancos de condensadores nos barramentos da rede elétrica em estudo, a segunda é a capacidade desses bancos de condensadores. O que se pretende é analisar como é afetada a rede elétrica, ou seja, se as restrições físicas da rede são respeitadas e se a qualidade de serviço é melhorada posteriormente à implementação de banco de condensadores.

Caso a implementação de um banco de condensadores seja apenas feita num barramento da rede elétrica, é calculado o fluxo de potências da rede e aumentada a capacidade do banco até as perdas serem mínimas. No entanto, caso a implementação de bancos de condensadores seja feita em dois ou mais barramentos da rede elétrica, o cálculo de potências já será feito para várias combinações possíveis, respeitando sempre as restrições da rede, até que seja encontrada a combinação ideal, aquela que irá minimizar ou maximizar a função objetivo. A complexidade do problema consiste no número de barramentos da rede, quanto maior for o número de barramentos maior será a complexidade, originando um maior número de combinações possíveis. Assim, estamos perante um problema de natureza combinatória. O problema também apresenta uma natureza não linear, devido às perdas por efeito de Joule.

As variáveis de decisão do problema serão a capacidade do banco de condensadores a implementar em cada barramento. A capacidade disponível no mercado de um condensador é limitada a certos

valores. Deste modo, a variável decisão do problema seria discreta. Para facilitar a resolução do problema *OCP*, a variável de decisão do problema será contínua, mas poderá apenas tomar valores inteiros.

2.1.3 Técnicas e algoritmos de otimização para o problema

Uma vez que o problema *OCP* é um problema de otimização combinatória, têm sido propostas, no decorrer do tempo, variadas técnicas e algoritmos de otimização. De seguida são apresentadas essas técnicas e algoritmos, constituídas por: métodos analíticos, métodos de programação matemática, métodos heurísticos e métodos de inteligência artificial.

Métodos analíticos

Na década de 1950, quando os recursos computacionais potentes não eram disponíveis ou eram caros, os autores dos métodos analíticos serviram-se dos cálculos das primeiras pesquisas e estudos acerca de colocação de condensadores para minimização de perdas numa rede elétrica. Estes métodos eram baseados em suposições como cargas uniformes, capacidade de condensadores não discretos, capacidades iguais de condensadores e locais de condensadores constantes.

A regra dos dois terços surgiu a partir destes métodos analíticos, desenvolvida por Schmill[4]. De acordo com a regra dos dois terços, para cargas uniformemente distribuídas, a capacidade do condensador ideal deve ser dois terços dos exigidos pelo circuito. A localização ótima deste condensador será a dois terços da subestação até ao fim da linha. Através desta localização do condensador, para cargas uniformemente distribuídas, a fonte subestação fornece potência reativa para o primeiro terço do circuito, e o condensador fornece potência reativa para os últimos dois terços do circuito, como se pode ver pela Figura 2.2 [4].

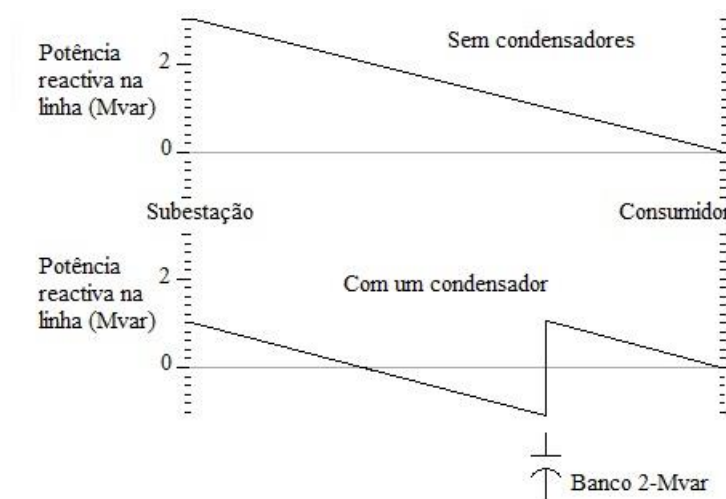


Figura 2.2: Condensador ideal para redução de perdas usando a regra dos dois terços.

Os métodos analíticos iniciais para a implementação de condensadores foram desenvolvidos sobretudo por Neagle e Samson [5]. O problema é definido para determinar a localização e tamanho de um dado número de condensadores do tipo fixo com o intuito de minimizar as perdas energéticas num determinado nível de carga.

Métodos de programação matemática

A programação matemática é uma técnica onde os problemas matemáticos são formulados de modo a que possam ser resolvidos através de operações aritméticas. Estes métodos podem ser considerados como técnicas iterativas, que têm a finalidade de maximizar ou minimizar uma função objetivo contendo variáveis de decisão. Para o problema *OCP*, as funções de perdas ou de poupanças serão as funções objetivo. As variáveis de decisão são as capacidades dos condensadores e o número de condensadores. Os valores destas variáveis devem satisfazer um conjunto de restrições.

No problema *OCP* foram formulados diferentes modelos matemáticos, que utilizaram métodos numéricos para resolver o problema. Duran [6] usou a abordagem de programação dinâmica e implementou o trabalho de Schmill [4], com cargas uniforme e aleatoriamente distribuídas, para encontrar o local ótimo para implementação de condensadores. A proposta de Duran era simples e apenas interessava a redução de perdas de energia, considerando apenas tamanhos de condensadores discretos.

Métodos heurísticos

Métodos heurísticos são desenvolvidos a partir da intuição, da experiência e da avaliação. Estes métodos são considerados estratégias rápidas e práticas que têm a finalidade de reduzir o espaço de busca, originando soluções eficientes próximas da solução ideal. Os resultados produzidos por algoritmos heurísticos não são garantidamente soluções ideais. Estes métodos também costumam ser associados a problemas de *OCP*.

Salam *et al.* [7] apresentaram uma estratégia heurística onde variava as cargas para reduzir as perdas do sistema, identificando, assim, os nós mais significativos em que os condensadores deviam ser instalados. A capacidade do condensador deve ser a que exibe menores perdas elétricas para a rede, contando sempre com as restrições das tensões nos barramentos. Chris *et al.* [8] estenderam o trabalho de Salam e consideraram o custo do banco de condensadores (capacidade) para minimizar perdas de energia e perdas de pico de potência.

2.1.3.4 Métodos de inteligência artificial

Os métodos heurísticos, apesar de simples, são algoritmos de busca que procuram todas as soluções possíveis preestabelecidas e, por conseguinte, escolhem a melhor solução. Ainda assim, estes métodos não são tão eficientes em tempo de cálculo e necessidade de espaço. Métodos de inteligência artificial

têm sido os algoritmos mais utilizados nos últimos anos para resolver problemas complexos no domínio da investigação operacional. Estes algoritmos, também considerados como meta heurísticas, são uma classe especial dos métodos heurísticos de pesquisa. São métodos de otimização baseados na inteligência que têm sido muito utilizados para encontrar a melhor localização e capacidade de condensadores numa rede elétrica.

Vários algoritmos foram utilizados para resolver este problema de otimização, como por exemplo, Algoritmos Genéticos [9], Pesquisa Tabu [10], Otimização por Enxame de Partículas [9] [10], Otimização por Colônia de Formigas [13], Arrefecimento Simulado [14], entre outros. No entanto, em redes de grandes escalas, estes métodos podem sofrer de elevados tempos de cálculo e grande requisito de espaço de memória, em tais casos, os métodos híbridos[15] podem ser uma excelente opção. Os métodos híbridos são métodos que combinam dois ou mais algoritmos para resolver um problema. Geralmente combinam as melhores características de cada algoritmo de modo que o algoritmo global seja melhor que os algoritmos individuais.

2.2 Formulação do problema

A resolução do problema visa determinar uma solução de estado estacionário viável de um sistema de energia elétrica envolvendo restrições físicas e operacionais, minimizando a função objetivo responsável por minimizar as perdas e melhorar o perfil de tensão.

O modelo matemático do problema *OCP* em sistemas de distribuição pode ser expresso da seguinte forma:

$$\min \text{Perdas}(\text{localização}, \text{capacidade}) \quad 2.3$$

Sujeito a,

$$V_{\min} \leq |V_i| \leq V_{\max} \quad 2.4$$

$$|Q_i^c| \leq |Q_i^L| \quad 2.5$$

Onde V_i é a tensão no barramento i , V_{\min} e V_{\max} são os limites de tensão mínimos e máximos, respetivamente; $|Q_i^c|$ (MVar) é o módulo da capacidade de um banco de condensadores no barramento i e $|Q_i^L|$ (MVar) é o módulo da potência reativa da carga no barramento i .

A função objetivo, 2.3, diz respeito unicamente às perdas elétricas numa rede elétrica. A tensão em cada barramento deve manter-se entre os limites mínimos e máximos de tensão. A capacidade de um banco de condensadores instalado, num determinado barramento, deve ser inferior à potência reativa da carga desse barramento, para evitar sobrecompensação.

Considerando o diagrama de apenas uma linha [16], Figura 2.3, o seguinte conjunto de equações recursivas é utilizado para o cálculo do fluxo de potência.

$$P_{i+1} = P_i - P_{i+1}^L - R_{i,i+1} \cdot \frac{P_i^2 + Q_i^2}{|V_i^2|} \quad 2.6$$

$$Q_{i+1} = Q_i - Q_{i+1}^L - X_{i,i+1} \cdot \frac{P_i^2 + Q_i^2}{|V_i^2|} \quad 2.7$$

$$|V_{i+1}^2| = |V_i^2| - 2(R_{i,i+1} \cdot P_i + X_{i,i+1} \cdot Q_i) + \frac{(R_{i,i+1}^2 + X_{i,i+1}^2)(P_i^2 + Q_i^2)}{|V_i^2|} \quad 2.8$$

Onde P_i e Q_i são as potências ativas e reativas que saem do barramento i , P_i^L e Q_i^L são as potências reais e reativas das cargas no barramento i . A resistência e reatância da secção da linha entre os barramentos i e $i + 1$ são representadas por $R_{i,i+1}$ e $X_{i,i+1}$, respetivamente.

As perdas da secção da linha entre os barramentos i e $i + 1$ podem ser calculadas a partir de:

$$P_{loss(i,i+1)} = R_{i,i+1} \cdot \frac{P_i^2 + Q_i^2}{|V_i^2|} \quad 2.9$$

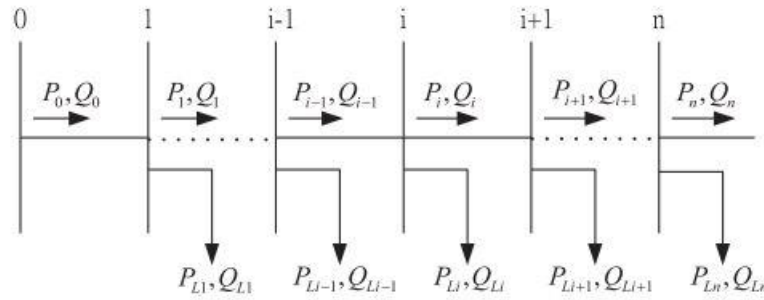


Figura 2.3: Diagrama de linha única de uma fonte principal.

As perdas totais, $P_{T,loss}$, podem, então, ser determinadas somando todas as perdas de todas as secções da linha.

$$P_{T,loss} = \sum_{i=0}^{n-1} P_{loss(i,i+1)} \quad 2.10$$

Assim, a função objetivo, *Perdas*, que se pretende minimizar, pode ser expressa por,

$$Perdas_{totais}(localização, capacidade) = P_{T,loss} \quad 2.11$$

2.3 Modelização

A modelização irá seguir os princípios da modelização matemática, uma representação de um sistema usando conceitos e linguagem matemática. Os modelos matemáticos podem tomar a forma de sistemas dinâmicos, modelos estáticos ou equações diferenciais.

No caso da modelização de redes elétricas, o principal objetivo é a identificação dos valores das tensões em todos os barramentos da rede, assim como das potências ativas e reativas que transitam em todas as linhas da rede elétrica. Para isso é necessário conhecer as condições de geração, de carga e a topologia da rede.

Na Figura 2.4 é exemplificada a modelização das linhas e barramentos de uma rede elétrica.

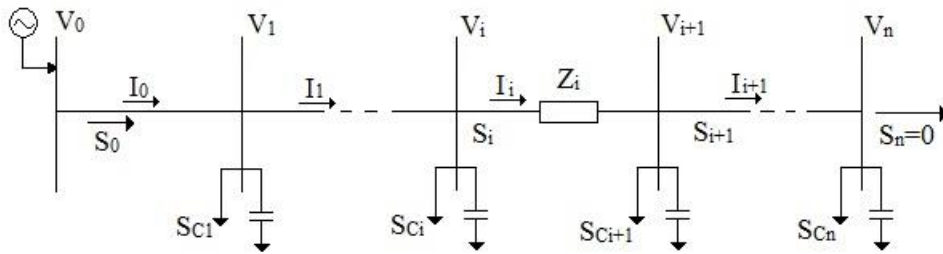


Figura 2.4: Modelização das linhas e barramentos de uma rede elétrica.

Considerando o barramento 0 como o de geração, a tensão V_0 é considerada constante. As linhas constituintes da rede elétrica são representadas por uma impedância $\bar{Z}_i = R_i + jX_i$ e as cargas são caracterizadas por potência constante $\bar{S}_{Ci} = P_{Ci} + jQ_{Ci}$, onde i é o número do barramento.

A potência injetada no barramento i é definida por $\bar{S}_i = \bar{S}_{Gi} - \bar{S}_{Ci}$, onde \bar{S}_{Gi} é a potência gerada no barramento i .

2.3.1 Trânsito de energia

Correntes injetadas nos barramentos

Numa rede elétrica de n barramentos, aplicando a 1ª lei de *Kirchoff* a cada barramento, obtém-se:

$$\begin{bmatrix} \bar{I}_0 \\ \bar{I}_1 \\ \vdots \\ \bar{I}_n \end{bmatrix} = \begin{bmatrix} \bar{Y}_{0,0} & \bar{Y}_{0,1} & \dots & \bar{Y}_{0,m} \\ \bar{Y}_{1,0} & \bar{Y}_{1,1} & \dots & \bar{Y}_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{Y}_{m,0} & \bar{Y}_{m,1} & \dots & \bar{Y}_{m,m} \end{bmatrix} \begin{bmatrix} \bar{I}_0 \\ \bar{I}_0 \\ \vdots \\ \bar{I}_0 \end{bmatrix} \quad 2.12$$

Que também pode ser expresso da seguinte maneira,

$$[\bar{I}] = [\bar{Y}][\bar{V}] \quad 2.13$$

onde \bar{I} é o vetor de correntes injetadas, $[\bar{Y}]$ é a matriz de admitâncias nodais e \bar{V} é o vetor das tensões nos barramentos.

Potências injetadas num barramento

Cada barramento pode ser caracterizado por quatro variáveis, sendo elas:

- Potência ativa concedida à rede e/ou a uma carga;
- Potência reativa concedida à rede e/ou a uma carga;
- Amplitude da tensão no barramento;
- Ângulo da tensão no barramento.

A potência S_i injetada num barramento i pode ser representada pela corrente injetada, I_i , nesse barramento à tensão V_i , considerando todas as variáveis na unidade pu .

$$\bar{S}_i = P_i + jQ_i = \bar{V}_i \bar{I}_i^* \text{ ou } \bar{S}_i^* = P_i - jQ_i = \bar{V}_i^* \bar{I}_i \quad 2.14$$

Considerando x como o número total de nós da rede e j o barramento a jusante do barramento i , obtém-se a seguinte expressão,

$$\bar{I}_i = \sum_{j=1}^x \bar{Y}_{i,j} \bar{V}_j \quad 2.15$$

Substituindo na expressão 2.14,

$$P_i - jQ_i = \bar{V}_i^* \sum_{j=1}^x \bar{Y}_{i,j} \bar{V}_j \quad 2.16$$

Onde se pode definir \bar{V}_m , \bar{V}_k e $\bar{Y}_{m,k}$ através da amplitude e fase,

$$\bar{V}_i = |V_i| \angle \delta_i \quad 2.17$$

$$\bar{V}_j = |V_j| \angle \delta_j \quad 2.18$$

$$\bar{Y}_{i,j} = |Y_{i,j}| \angle \theta_{i,j} \quad 2.19$$

Alterando assim a expressão 2.16 para,

$$P_i - jQ_i = \sum_{j=1}^x |V_i| |V_j| |Y_{i,j}| \angle (-\delta_i + \delta_j + \theta_{i,j}) \quad 2.20$$

Separando a parte real da imaginária,

$$P_i = \sum_{j=1}^x |V_i| |V_j| |Y_{i,j}| \cos(-\delta_i + \delta_j + \theta_{i,j}) \quad 2.21$$

$$Q_i = \sum_{j=1}^x |V_i| |V_j| |Y_{i,j}| \sin(-\delta_i + \delta_j + \theta_{i,j}) \quad 2.22$$

As duas expressões, 2.21 e 2.22, representam as potências ativa e reativa, referentes ao barramento i (a energia transita do barramento i para o j).

Potências transitadas nas linhas

Na Figura 2.5 está representada uma linha r , que liga os barramentos i e j . Para se determinar a potência ativa e reativa, que transitam na linha, é necessário saber a amplitude e a fase das tensões de ambos os barramentos.

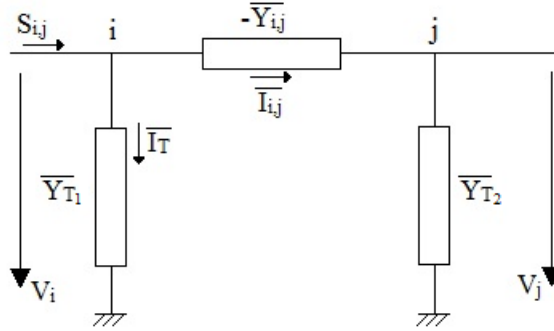


Figura 2.5: Potência transitada numa linha

O fluxo de potência que transita do barramento i para o barramento j é expresso por,

$$\overline{S}_{l,j} = \overline{V}_i (\overline{I}_r^* + \overline{I}_{l,j}^*) \text{ ou } \overline{S}_{l,j}^* = \overline{V}_i^* (\overline{I}_r + \overline{I}_{l,j}) \quad 2.23$$

Considerando que,

$$\overline{I}_r = \overline{Y}_T \overline{V}_i \quad 2.24$$

$$\overline{I}_{l,j} = -\overline{Y}_{l,j} (\overline{V}_i - \overline{V}_j) \quad 2.25$$

Onde I_T é a corrente no ramo transversal e $I_{l,j}$ a corrente no ramo longitudinal. $Y_{i,j}$ é um elemento da matriz de admitâncias nodais, que, como se pode observar, é o simétrico da admitância longitudinal da linha que liga o barramento i ao j .

Substituindo na expressão 2.23, obtém-se a potência complexa transmitida do barramento i para o j ,

$$\overline{S_{i,j}}^* = (Y_T - Y_{i,j})V_i^2 + Y_{i,j} V_i^* V_j$$

2.26

2.3.2 Cálculo das equações do trânsito de energia

Barramentos

Numa rede elétrica podem encontrar-se três tipos de barramentos:

- Barramento de referência ou balanço: neste barramento tanto a amplitude como a fase da tensão são conhecidas. Geralmente o ângulo é considerado igual a zero.
- Barramentos de carga ou PQ: nestes barramentos são ligadas apenas cargas, sendo que as potências injetadas nesse barramento pelas cargas são negativas (consomem energia da rede). As potências ativa e reativa da carga são conhecidas.
- Barramento de geração ou PV: nestes barramentos são ligados os geradores e também é possível ligar cargas. Tanto a potência ativa gerada pelo gerador e a amplitude da tensão são conhecidas.

Na Tabela 2 são apresentadas as variáveis que se conhecem, as variáveis que são especificadas e as que precisam de ser calculadas, consoante o tipo de barramento. As potências ativa e reativa da carga são representadas por P_C e Q_C , respetivamente; as potências ativa e reativa geradas no barramento são representadas por P_G e Q_G , respetivamente; $|V|$ significa a amplitude e o θ a fase da tensão no barramento.

Tabela 2: Tipos de barramentos.

Tipo de barramento	Variáveis conhecidas		Variáveis especificadas		Variáveis calculadas	
Referência	P_C	Q_C	$ V $	θ	P_G	Q_G
PQ	P_C	Q_C	P_G	Q_G	$ V $	θ
PV	P_C	Q_C	P_G	$ V $	Q_G	θ

Linhas

O modelo adotado como esquema equivalente da linha de transmissão de energia é o presente na Figura 2.6, onde são ilustrados os parâmetros necessários para os cálculos e se pode ver a impedância longitudinal $\overline{Z}_L = R_L + jX_L$.

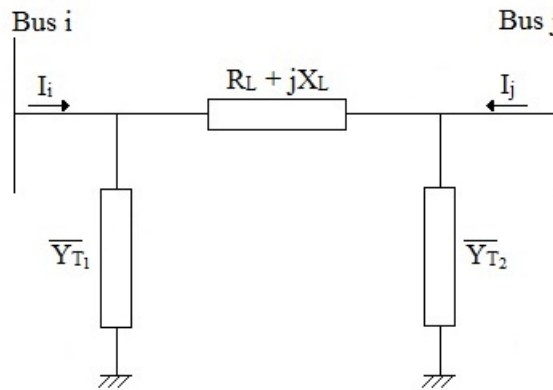


Figura 2.6: Esquema equivalente em π de uma linha de transmissão.

Método aplicado para o cálculo do trânsito de energia

A solução das equações do trânsito de energia passa por calcular as tensões (amplitude e fase) nos barramentos da rede. Devido à não linearidade das equações do trânsito de energia, tem de se utilizar um processo iterativo. O método adotado para calcular essas equações foi o método de *Newton-Raphson*.

O método de *Newton-Raphson*, como outros processos iterativos, segue uma certa sequência de processos. No início, é estimado um valor para as amplitudes e fases das tensões em todos os barramentos. Depois de calculadas as correções para esses valores, somam-se essas correções aos valores inicialmente estimados. Posteriormente é feita uma comparação entre as amplitudes das tensões com as condições de exatidão estipuladas, para todos os barramentos. Quando estas condições forem cumpridas, o algoritmo termina. A solução encontrada será tanto mais precisa quanto maior for o número de iterações. Contudo, um número elevado de iterações pode levar a um ciclo infinito, caso o processo divirja.

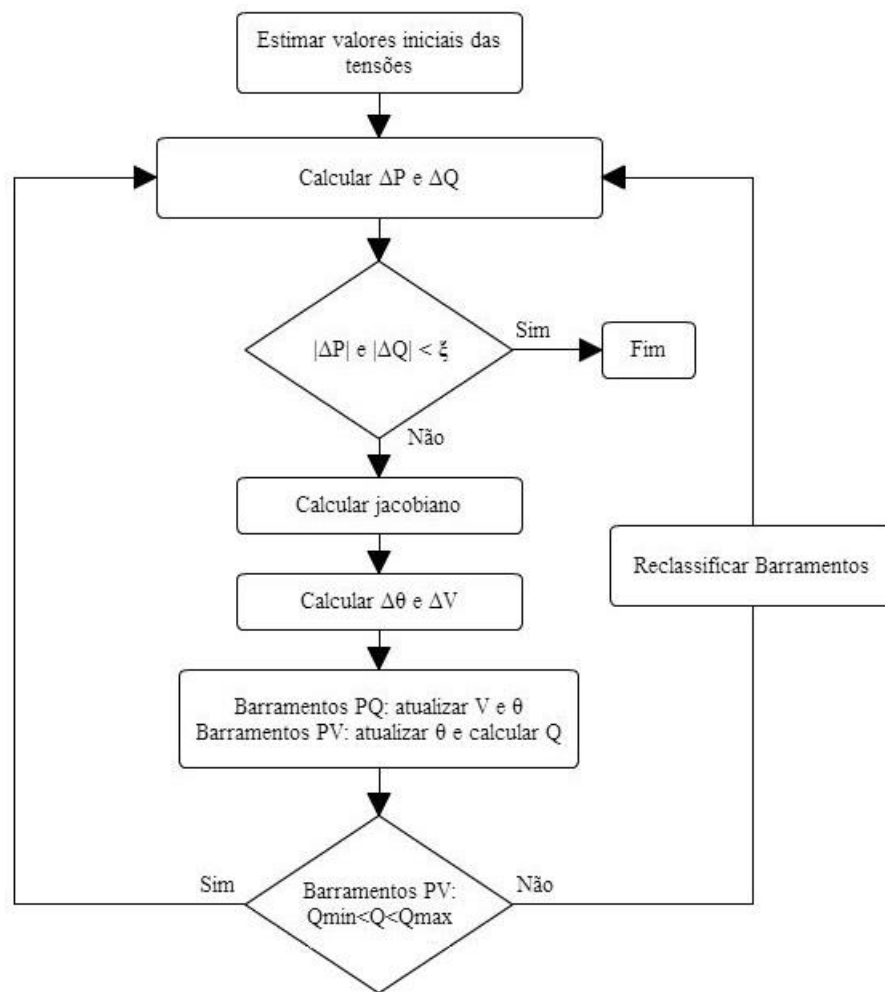


Figura 2.7: Fluxograma do método Newton-Raphson [17].

3. Metodologias de otimização

Apresentação das bases
dos algoritmos aplicados ao
problema *OCP*. Descrição dos algoritmos.

3.1 Meta-heurística

O termo meta-heurística deriva da composição de duas palavras gregas: heurística deriva do verbo *heuriskein* (εὕρισκειν) que significa "encontrar", enquanto o sufixo meta significa "além de, em um nível superior". Antes de este termo ter sido adotado, a meta-heurística foi muitas vezes chamada de heurística moderna [18] .

Meta-heurística são estruturas algorítmicas gerais, muitas vezes inspiradas na natureza, projetadas para resolver problemas de otimização complexos, e tem sido uma área crescente de investigação nas últimas décadas. Nos últimos anos, a meta-heurística tem surgido como alternativa de sucesso para abordagens mais clássicas para resolver problemas de otimização [19] .

De acordo com Blum e Roli [18], as propriedades fundamentais que caracterizam a meta-heurística são:

- Meta-heurística são estratégias que "guiam" o processo de pesquisa;
- O objetivo é explorar de forma eficiente o espaço de busca, a fim de encontrar soluções sub ótimas;
- Técnicas que constituem algoritmos meta-heurísticos variam de simples procedimentos de busca local para processos de aprendizagem complexos;
- Algoritmos meta-heurísticos são aproximados e geralmente não-deterministas;
- Podem incorporar mecanismos para evitar ficar preso em áreas do espaço de busca;
- A meta-heurística mais avançada utiliza experiência de pesquisa para orientar a busca.

Várias ferramentas da meta-heurística tem evoluído, no decorrer das últimas duas décadas, com o intuito de facilitar a solução de problemas de otimização que antes eram difíceis de resolver. Algumas dessas ferramentas são, Algoritmos Genéticos (*Genetic Algorithm*), Estratégias de Evolução e Programação Evolucionária (*Evolution Strategies and Evolutionary Programming*), Otimização por Enxame de Partículas (*Particle Swarm Optimization*), Pesquisa Tabu (*Tabu Search*), Arrefecimento Simulado (*Simulated Annealing*).

Neste estudo iremos abordar o algoritmo *PSO* e um algoritmo proposto por Vladimiro Miranda [20], que consiste num algoritmo híbrido que junta o mundo das Estratégias de Evolução com o da *PSO*, denominado de *EPSO* (*Evolutionary Particle swarm optimization*).

3.2 Computação Evolucionária

Porquê usar a evolução como inspiração para a resolução de problemas computacionais? Os mecanismos de evolução parecem bem adequados para alguns dos problemas computacionais mais urgentes em muitos campos. Muitos problemas computacionais envolvem a busca através de um grande número de possibilidades de soluções. Nas décadas de 1950 e 1960, vários cientistas de computação investigaram a possibilidade de usarem sistemas evolutivos como uma ferramenta de otimização para problemas da engenharia. A ideia principal nestes sistemas era evoluir uma população de soluções candidatas a um determinado problema, evolução que era feita a partir de operadores inspirados na variação da genética natural e na seleção natural [21] .

Na década de 1960, na Universidade Técnica de Berlim, Rechenberg começou a formular ideias acerca de como poderia resolver problemas de otimização de valores reais com parâmetros difíceis, a partir de processos evolutivos. Assim, introduziu as "estratégias de evolução". Esta ideia foi desenvolvida por Schwefel (1975, 1977) que, hoje em dia, representa um dos mais fortes algoritmos evolucionários para problemas de otimização.

No mesmo período que surgiam as estratégias de evolução, na Universidade da Califórnia, em Los Angeles, Fogel, Owens, e Walsh viam o potencial de atingir as metas da inteligência artificial por meio de técnicas evolutivas. Esta ideia foi explorada, inicialmente, num contexto onde as soluções candidatas para determinadas funções eram representadas como máquinas de estados finitos, onde os seus diagramas de estado e transições evoluíam a partir de mutações aleatórias, sendo a solução mais apta selecionada. Esta ideia foi denominada por “programação evolucionária”, que, hoje em dia, é aplicada a uma ampla variedade de problemas, muito para além da evolução de máquinas de estados finitos.

Na década de 1960, John Holland criou os algoritmos genéticos e desenvolveu-os nessa mesma década e na seguinte, juntamente com os seus alunos e colegas da Universidade de Michigan. Holland viu nos processos evolutivos um elemento chave na conceção e implementação de sistemas adaptativos robustos que eram capazes de lidar com um ambiente incerto e de constante mudança. Em contraste com as estratégias de evolução e de programação evolucionária, o objetivo original de Holland não era projetar algoritmos para resolver problemas específicos, mas sim estudar o fenómeno da adaptação, comparado ao que ocorre na natureza, e desenvolver maneiras em que os mecanismos de adaptação natural possam ser importados em sistemas de computador. Estratégias de evolução, programação evolutiva e algoritmos genéticos integram os principais campos da computação evolucionária. Ambos os métodos de computação evolucionária seguem o ciclo da evolução, que é ilustrado pela Figura 3.1.

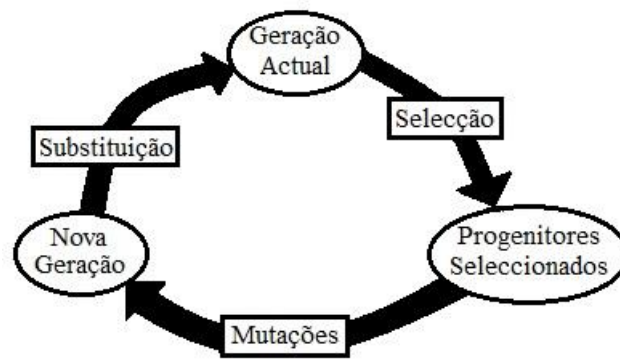


Figura 3.1: O ciclo básico da evolução [22].

3.2.1 Estratégias de Evolução e Programação Evolucionária

Estas duas variantes são designadas como métodos de fenótipo, o que significa que as soluções para um dado problema são representadas pelas próprias variáveis do problema, ou seja, o algoritmo não necessita de mecanismos externos para codificação/descodificação das soluções do problema. Estes métodos diferem dos métodos de genótipo, onde a solução de um dado problema é sempre codificada numa sequência (por exemplo, codificação binária), que necessitam de um algoritmo intermédio, capaz de explicitar as soluções em variáveis representativas do problema, ou vice-versa. Os métodos de genótipo representam os Algoritmos Genéticos.

Na *PE* (Programação Evolucionária) clássica, a variação (diferença entre os descendentes e progenitores) é feita a partir apenas de mutação, embora trabalhos precoces com máquinas de estados finitos já utilizassem também a recombinação para gerar novos descendentes. Nas *EE* (Estratégias de Evolução), processos idênticos à recombinação eram utilizados para gerar novos descendentes, apesar de inicialmente também terem sido utilizadas mutações para causar variação na população. Ambos os modelos têm vindo a convergir em termos conceptuais e, presentemente, são dificilmente distinguíveis.[23]

Estratégias de Evolução

As variáveis que compõem o fenótipo são as variáveis do problema. Outros parâmetros que existem nas *EE* são os parâmetros estratégicos, que estão relacionados com desvios padrão σ para as distribuições que são utilizadas nas probabilidades de mutação. Estas probabilidades podem ser globais ou diferentes para cada dimensão de um indivíduo.[23]

As primeiras *EE* eram baseadas em populações apenas de um indivíduo, Figura 3.2. Também só existia um operador genético usado para o processo de evolução, a mutação. O primeiro modelo a ser conhecido foi $(1+1)EE$. Este modelo significava que a seleção do progenitor da próxima geração seria selecionado entre o progenitor e o descendente. O descendente (o indivíduo mutado) é aceite na população se for mais apto que o seu progenitor e todas as condições (caso haja) forem satisfeitas, ou seja, o descendente substitui o progenitor. Por exemplo, sendo f uma função objetivo sem condições onde o objetivo é maximizá-la, um descendente x^{t+1} substitui o seu progenitor x^t se $f(x^{t+1}) > f(x^t)$. Caso contrário, o descendente é descartado e a população mantém-se inalterada. [24]

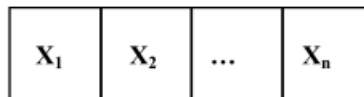


Figura 3.2: Representação de um indivíduo i com n variáveis reais.

Com o desenvolvimento das *EE*, novas estratégias apareciam, estas mais desenvolvidas e abrangentes. Outra notação era aplicada, surgiam as estratégias $(\mu, \lambda)EE$ e também a simples estratégia $(1+1)EE$ dava seguimento a novas estratégias do tipo $(\mu + \lambda)EE$. Nas estratégias $(\mu + \lambda)EE$, é selecionado para a nova geração os μ sobreviventes de entre o grupo da geração anterior, grupo que é constituído dos μ progenitores e seus λ descendentes. Nas estratégias $(\mu, \lambda)EE$, onde, obrigatoriamente, numa geração terá de existir mais descendentes que progenitores ($\lambda \geq \mu \geq 1$), a seleção dos μ progenitores da nova geração recai apenas sobre os λ descendentes da geração atual, independentemente da aptidão dos seus progenitores.

Usando a estratégia $(\mu, \lambda)EE$, um indivíduo apenas sobrevive uma geração, dando origem a outros indivíduos que, depois de selecionados, o irão substituir. Com a estratégia $(\mu + \lambda)EE$, um indivíduo, enquanto for mais apto que os seus descendentes, não será substituído para a próxima geração. É de notar que os modelos que seguem a estratégia $(.)$ permitem que o indivíduo mais apto, numa geração $t+1$, possa ser menos apto que o(s) indivíduo(s) de uma geração t . Este facto mostra que esta estratégia não é elitista, facilitando, assim, que sejam aceites deteriorações temporárias impossibilitando a atração ao ótimo local [24].

O algoritmo das *EE* pode ser caracterizado da seguinte forma, Figura 3.3.

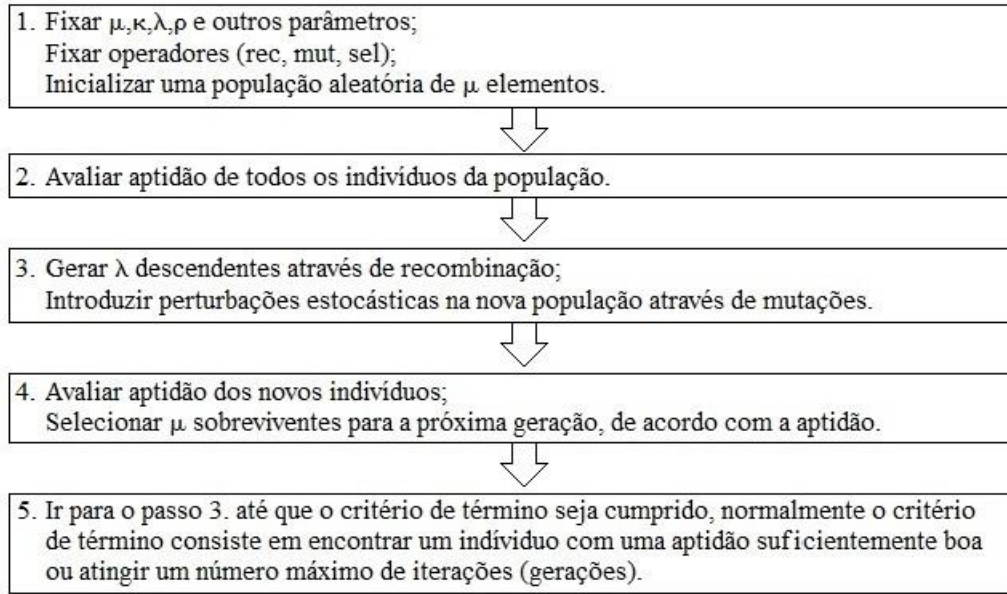


Figura 3.3: Algoritmo Estratégias de Evolução [23].

Operador Mutação e o Modelo Auto adaptativo

De acordo com Beyer e Schwefel [25], o operador mutação deve cumprir três condições:

- A primeira condição transmite que, de qualquer ponto no espaço de soluções, seja possível ter acesso a outro qualquer ponto, ou seja, as mutações introduzidas numa geração não podem fazer com que os indivíduos saiam do espaço de soluções;
- A segunda condição está relacionada com a tendência. Num espaço de soluções não restritas, a tendência é considerada uma desvantagem, visto que a direção para o ótimo é desconhecida, a existência de uma tendência pode levar a que o ótimo nunca seja alcançado;
- Por fim, a terceira condição, a “adaptação”, obriga o operador mutação a ser ajustável para adaptar a exploração e o desenvolvimento às condições locais do espaço de soluções.

A operação mutação pode ser entendida como uma variação numa geração g , variação essa que é traduzida pela adição ao indivíduo $X^{(g)}$ de uma perturbação Z , produzindo um novo indivíduo X^{g+1} .

$$X^{g+1} = X^g + Z \quad 3.1$$

O indivíduo X é um vector de n variáveis de decisão. A perturbação Z é feita de forma aleatória e é descrita na seguinte expressão,

$$Z = \sigma [N_1(0,1) \quad \cdots \quad N_n(0,1)]^t \quad 3.2$$

$N_j(0,1)$ corresponde a uma distribuição Gaussiana com média zero e variância unitária na variável de decisão j , e o desvio padrão σ é chamado de taxa de mutação.

Programação Evolucionária

Desde os seus primórdios da *PE*, o objetivo principal era simular a evolução para criar inteligência artificial[26]. Estas técnicas visavam a evolução da inteligência artificial no sentido de desenvolver a capacidade de prever as mudanças no ambiente. O ambiente era descrito como uma sequência de símbolos (a partir de um alfabeto finito) e o algoritmo de evolução produzia, como saída, um novo símbolo.

Embora a forma original da *PE* tenha sido proposta para operar em modelos de previsão, a maioria das atuais variantes da *PE* são utilizadas para problemas de otimização de parâmetros contínuos. Várias variantes da *PE*, nos tempos de hoje, são facilmente equiparadas às *EE*, devido às suas características muito semelhantes, tais como, a representação dos indivíduos e o operador de mutação. Como em qualquer método da computação evolutiva, num modelo típico de *PE*, é definida uma função de aptidão e uma população de indivíduos.

Atualmente, existe uma diferença entre *EE* e *PE*. Essa diferença está relacionada com o operador seleção, ao passo que na *EE* é adotado o método elitista (apenas os melhores indivíduos sobrevivem), na *PE* a seleção é feita através de torneios estocásticos (seleção aleatória com o auxílio de distribuições de probabilidade).

Na Figura 3.4, é apresentado um algoritmo que caracteriza o algoritmo da *PE*. Como se pode ver no passo 3, a variação da população é feita apenas pelo operador mutação.

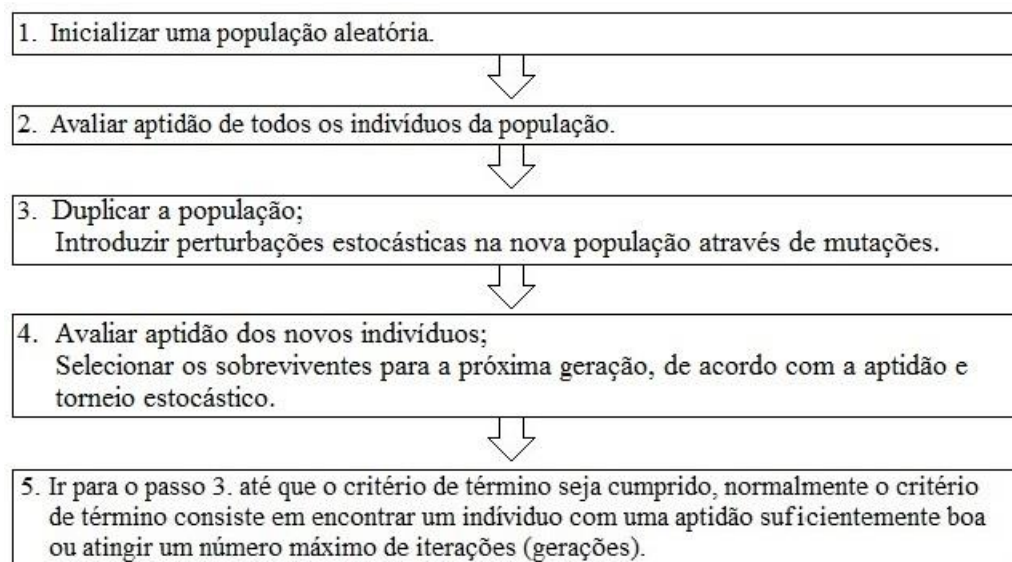


Figura 3.4: Algoritmo geral da Programação Evolucionária [23].

3.2.2 Algoritmos Genéticos

Os algoritmos genéticos (AG) foram desenvolvidos por John Holland[27]. Estes algoritmos são referenciados como genéticos devido à representação e manipulação dos indivíduos em termos da sua composição genética, ao invés de considerar uma representação fenotípica. A ideia principal dos AG é evoluir uma população de soluções candidatas, através de uma pressão seletiva, que favorece a sobrevivência do mais apto. AG é o algoritmo evolucionário mais conhecido para otimização de problemas. Usa técnicas inspiradas pela evolução natural, que permitem uma população de soluções (também chamados cromossomas), candidatas a resolver um problema de otimização, a evoluir, ou seja, a avançar em direção à melhor solução.

A evolução, normalmente, começa a partir de uma população de indivíduos aleatoriamente gerados, que mudam em cada iteração, chamada de geração. Em cada geração, a aptidão (os valores das funções objetivo) de cada cromossoma na população é avaliada, e os cromossomas mais adequados (aqueles com maiores ou menores valores de aptidão, dependendo se a finalidade é a maximização ou minimização, respetivamente) são selecionados. Cada cromossoma é modificado por meio de cruzamentos e uma nova, mais evoluída, geração de soluções de candidatos é formada. Os cromossomas em uma população AG frequentemente assumem a forma de sequências de bits (*strings* de 1 e 0); ou seja, cada posição de bit no cromossoma tem dois possíveis valores, 0 e 1. No exemplo [28], da Figura 3.5, pode observar-se a aplicação de um AG num problema de navegação robótica (movimentos "para frente", "para atrás", "esquerda", e "direita", comandos de paragem emergência são desprezados), em que uma solução candidata é codificada como uma *string* de *bits*.

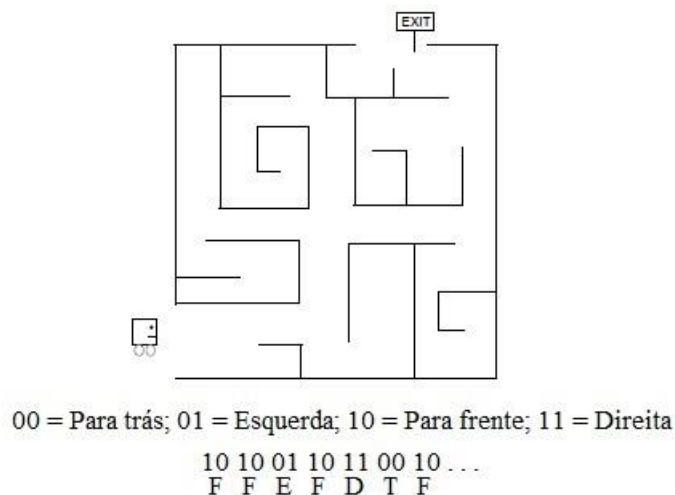


Figura 3.5: Aplicação de um AG num problema de navegação robótica [28].

A maioria dos métodos dos AG tem, pelo menos, os seguintes elementos em comum: população de cromossomas, seleção de acordo com a aptidão, cruzamento de cromossomas para produzir novos descendentes e mutações aleatórias nos novos descendentes.

De acordo com M. Mitchell[28], os três operadores (seleção, cruzamento e mutação) podem ser definidos, em relação ao problema de navegação robótica, da seguinte maneira:

- Seleção: este operador seleciona cromossomas na população para a reprodução. Quanto mais apto for o cromossoma, mais provável será a sua seleção.
- Cruzamento: este operador troca subsequências de dois cromossomas para criar dois descendentes. Por exemplo, as *strings*

10000100 e 11111111

podem ser cruzadas a partir da terceira posição para produzir os dois descendentes

10011111 e 11100100.

- Mutação: Este operador inverte aleatoriamente alguns bits de um cromossoma. Por exemplo, a *string* 0000010 pode ser mutada, na sua segunda posição, obtendo-se 0100010. A mutação pode ocorrer em cada posição de bit de uma *string* com alguma probabilidade, geralmente muito pequena (por exemplo, 0.001) [28].

O AG pode ser exemplificado pelo seguinte fluxograma, Figura 3.6.

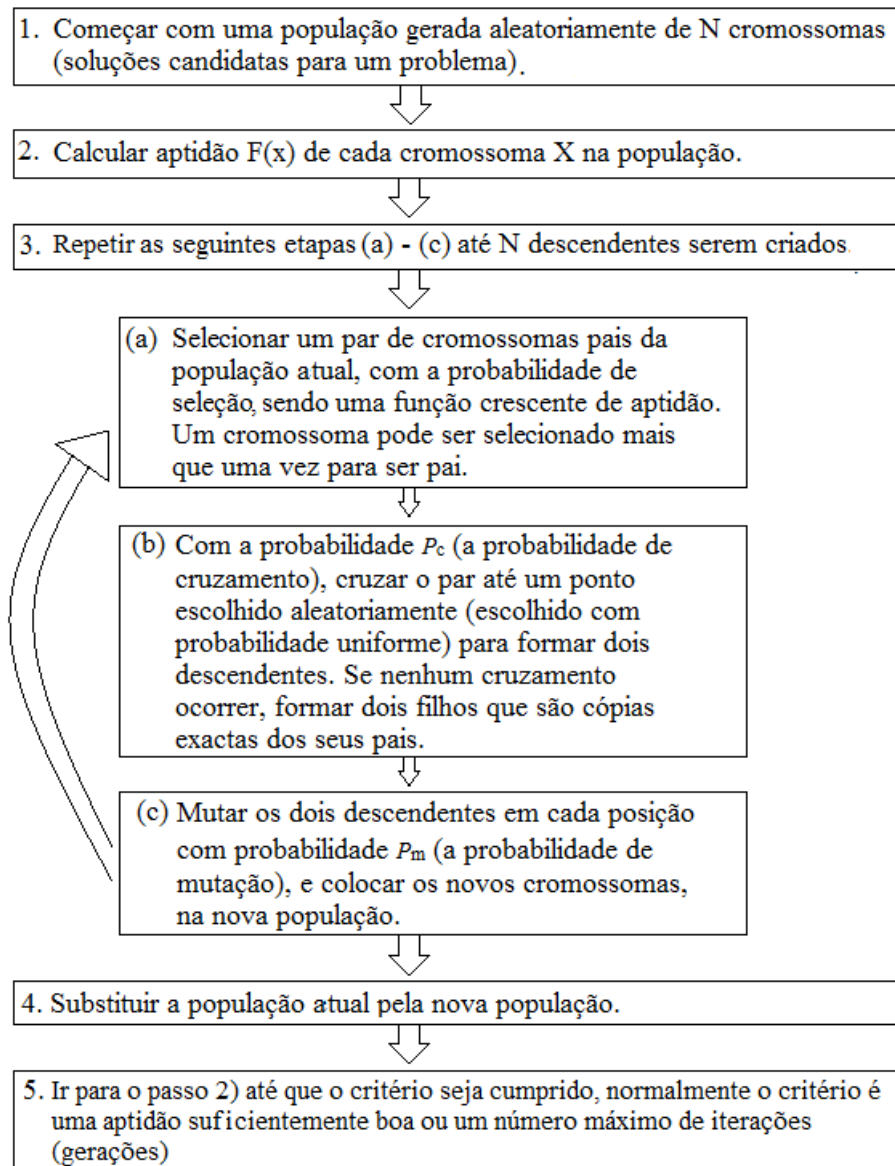


Figura 3.6: Fluxograma que exemplifica o funcionamento do Algoritmo Genético [28] .

Quando uma geração é criada, é expectável que um ou mais cromossomas sejam altamente aptos. Apesar deste facto, o comportamento do algoritmo depende da sua correta codificação e parametrização. Há um número de parâmetros que se tem de considerar, tais como o tamanho da população e as probabilidades de cruzamento e mutação. O sucesso do algoritmo depende muitas vezes destes parâmetros.

3.3 Otimização por Enxame de Partículas

A otimização por enxame de partículas (*PSO*) foi originalmente desenvolvida por James Kennedy e Russel Eberhart em 1995, e tem sido comparada com algoritmos genéticos [38], [39], [40] pela eficiência em problemas de otimização.

O socio biólogo, E. O. Wilson [32], escreveu, em relação aos cardumes de peixes, “Pelo menos em teoria, os membros individuais do cardume podem tirar proveito das descobertas e experiências anteriores de todos os outros membros do cardume durante a busca de alimentos. Esta vantagem pode se tornar determinante, superando as desvantagens da concorrência na busca de alimentos, quando o recurso é distribuído de forma imprevisível e em poucas quantidades”. Esta declaração transmite que a partilha social de informação entre espécies oferece uma vantagem evolutiva. Esta suposição foi fundamental para o desenvolvimento da *PSO* [33].

3.3.1 Objetivo da *PSO*

O objetivo inicial deste algoritmo era simular graficamente a coreografia elegante, mas imprevisível, de um bando de pássaros ou de um cardume de peixes. Anteriormente ao aparecimento da *PSO*, haviam sido apresentadas simulações de bandos de pássaros por cientistas como Reynolds [34] e Heppner e Grenander [35]. Ambos os modelos baseavam-se na manipulação das distâncias entre indivíduos, ou seja, a sincronia do comportamento do bando era traduzida no esforço que os pássaros faziam para manterem a distância perfeita entre si e os seus vizinhos. [33]

A *PSO* é baseada em duas ciências distintas, a ciência social e a ciência da computação:

- Conceitos Sociais: Avaliação, comparação e imitação de outros, bem como a aprendizagem a partir da experiência, permitem que os seres humanos se adaptem ao ambiente e determinem os padrões ideais de comportamento, atitudes, entre outros;
- Inteligência Computacional: A importância do conceito de inteligência computacional é a adaptação do sistema que permite ou facilita um comportamento inteligente em ambientes complexos de constante mudança.

3.3.2 Descrição do algoritmo

O principal conceito da *PSO* é a partilha de informação e colaboração entre indivíduos (conjuntos de soluções que evoluem no espaço de alternativas – *partículas*) através de simulação do seu comportamento social. Cada partícula i é considerada como uma potencial solução para um dado problema de otimização e corresponde a um ponto num espaço D-dimensional. Uma partícula é representada, então, pelos seguintes vetores:

- Vetor de Posição $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$
- Vetor de Velocidade $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$

- Vetor para a melhor posição ocupada pela partícula até ao momento, $b_i = (b_{i1}, b_{i2}, \dots, b_{iD})$

A melhor posição encontrada pelo conjunto total de partículas (enxame) é memorizada no vetor $b_G = (b_{G1}, b_{G2}, \dots, b_{GD})$.

A posição da partícula i num instante t [30] é dada por 3.3:

$$X_i(t) = X_i(t - 1) + V_i(t) \quad 3.3$$

A informação a que cada partícula tem acesso é baseada na sua experiencia (as decisões que fez e o sucesso das mesmas) e no conhecimento do desempenho das outras partículas do enxame. Como a importância destes dois fatores pode variar de decisão para decisão, pesos aleatórios são aplicados a estes dois fatores [30]. Assim a equação da velocidade no modelo original é dada pela expressão 3.4:

$$V_i(t) = V_i(t - 1) + c_1 \cdot rand_1 \cdot (b_i - X_i) + c_2 \cdot rand_2 \cdot (b_G - X_i) \quad 3.4$$

Sendo que c_1 e c_2 são duas constantes, fixadas externamente, que tomam valores positivos, $rand_1()$ e $rand_2()$ são dois números aleatórios com distribuição uniforme num intervalo $[0.0, 1.0]$.

Na Figura 3.7 é ilustrado o movimento da partícula, que é dividido em três componentes [36]:

1. A primeira componente é muitas vezes chamada de “inércia” ou “momentum”. Esta componente fará com que a partícula tenha tendência de continuar sempre na mesma direção.
2. A segunda componente é uma atração linear para a melhor posição já encontrada pela partícula. Esta componente é a componente cognitiva e pode tomar nomes como “memória”, “nostalgia” ou “auto conhecimento”.
3. A terceira componente é uma atração linear para a melhor posição já encontrada por qualquer partícula. Esta componente é a componente social e pode tomar nomes como “cooperação”, “conhecimento social” ou “informação partilhada”.

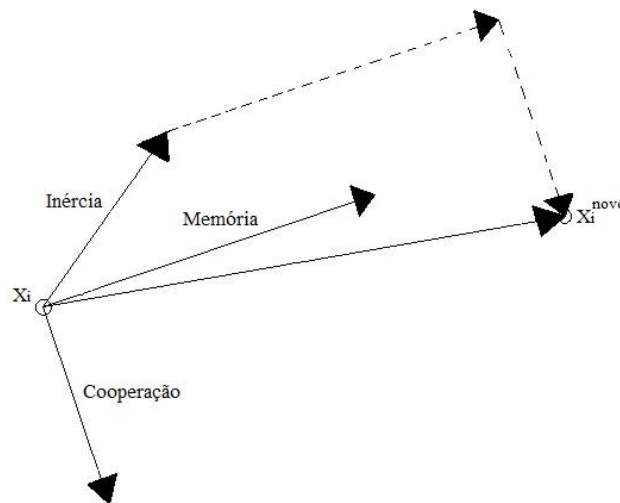


Figura 3.7: Ilustração das três componentes do movimento de uma partícula i .

Num modelo modificado [37], e o mais corrente nos tempos de hoje, foi introduzido uma constante w , fixada externamente, que irá balancear a procura global e a procura local. Analisando a expressão 3.4, é perceptível que, sem a primeira componente da expressão, todas as partículas se iriam mover em direção à mesma posição. A equação da velocidade do modelo modificado é, então, a seguinte:

$$V_i(t) = w \cdot V_i(t-1) + c_1 \cdot rand_1 \cdot (b_i - X_i) + c_2 \cdot rand_2 \cdot (b_G - X_i) \quad 3.5$$

A velocidade da partícula em cada dimensão está limitada por uma velocidade máxima, V_{\max} e uma velocidade mínima, V_{\min} . Portanto, V_{\max} é considerado um parâmetro importante. Este parâmetro irá ilustrar a distância que a partícula se poderá mover numa determinada dimensão. Caso V_{\max} seja muito grande, a partícula poderá passar por uma boa solução, por outro lado, caso V_{\max} seja muito pequeno, a partícula poderá não explorar para além de um certa região ou ótimo local.

Outro aspeto importante no algoritmo da *PSO* é a topologia do enxame. Foram feitos estudos, seguindo duas redes distintas de vizinhança: melhor global e o melhor local. Na rede melhor global, a partícula é atraída para a melhor solução encontrada por qualquer partícula do enxame. Cada partícula tem acesso à informação de todas as outras partículas, Figura 3.8(a) [38]. Porém, na rede melhor local, cada partícula tem apenas acesso à informação do seu vizinho mais próximo, de acordo com a topologia escolhida. As topologias mais comuns do tipo de rede melhor local são a topologia em anel, onde cada partícula está ligada a dois vizinhos, Figura 3.8(b) [38], e a topologia radial, onde cada partícula apenas comunica com uma partícula focal, sendo que a partícula focal comunica com todas as outras restantes, Figura 3.8(c) [38].

A rede melhor global converge mais rápido, mas a melhor solução encontrada pode ficar presa em um mínimo local (no caso do objetivo da otimização seja minimizar), ao passo que a rede melhor local tem melhores oportunidades de achar a solução ótima, embora convirja lentamente.[39]

Kennedy and Mendes [40] avaliaram todas as topologias presentes na Figura 3.8 [38], assim como o caso de vizinhos aleatórios. Nesse estudo, com um número total de vinte partículas, concluíram que o melhor desempenho ocorria na geração aleatória de uma vizinhança com um tamanho médio de cinco partículas. No entanto, a seleção da rede de vizinhança mais eficiente, em geral, depende do tipo de problema. Uma estrutura pode ser mais eficiente para certos tipos de problemas, todavia pode ter um desempenho pior para outros problemas.

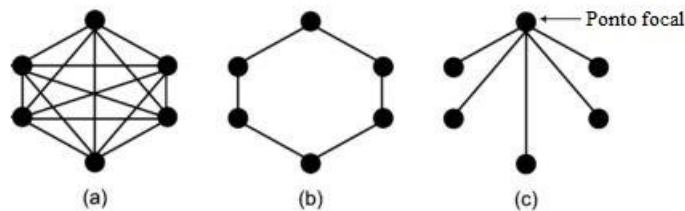


Figura 3.8: Topologias de Enxame. a) Melhor global. b) Topologia em anel. c) Topologia radial .

O algoritmo da *PSO* pode ser elucidado pelo seguinte fluxograma, Figura 3.9.

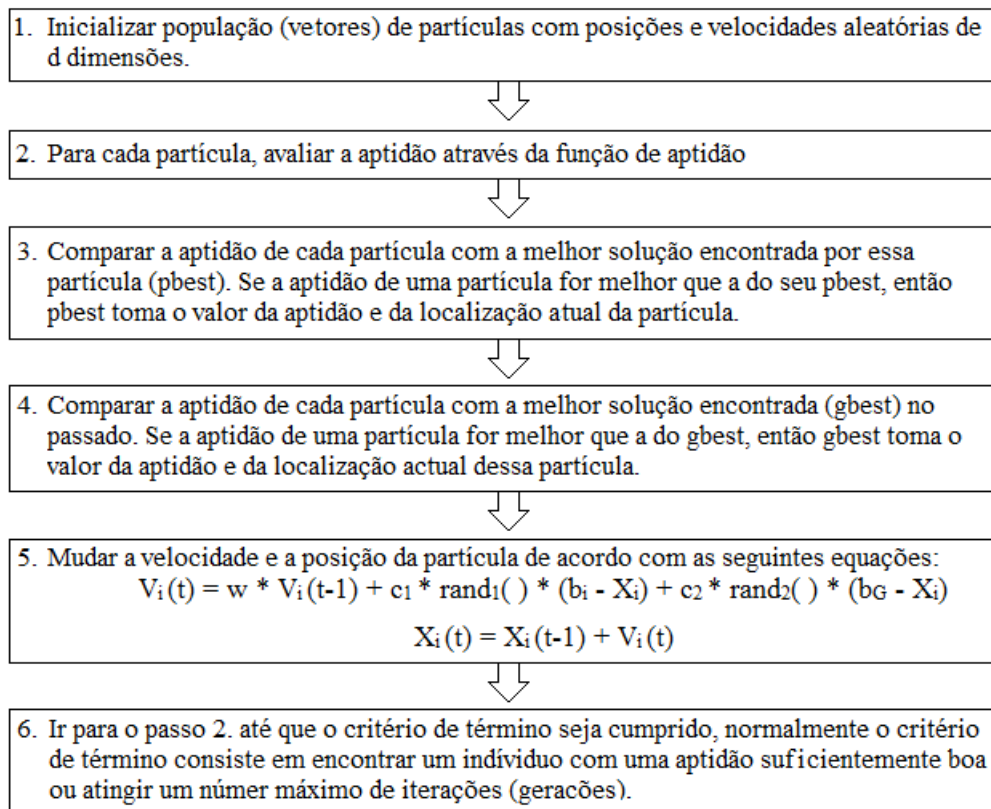


Figura 3.9: Fluxograma que exemplifica o funcionamento da *PSO*.

3.4 Otimização por Enxame de Partículas Evolucionário (*EPSO*)

Como foi referido na secção 3.1, um dos algoritmos que irá ser analisado é o *EPSO*, que surge como um algoritmo híbrido pois contempla os dois mundos distintos, estratégias evolutivas e otimização por enxame de partículas. Nos modelos autoadaptativos (AA) das *EE*, σ AA-ES, onde tanto os indivíduos evoluem em direcção a uma solução ótima, bem como os seus parâmetros estratégicos responsáveis pelo operador de mutação (σ). Este acontecimento transmite um processo de auto aprendizagem que resulta numa melhor forma de progressão para o ótimo. Na *PSO*, não existe competição entre partículas nem autoadaptação das suas características (pesos). É certo que, caso não existisse a componente “cooperação”, cada partícula evoluiria independentemente [41]. De facto, no algoritmo da *PSO*, é a equação de movimento e esta componente de cooperação que garantem as propriedades de convergência e o funcionamento do algoritmo.

3.4.1 Objetivo da *EPSO* e suas origens

O *EPSO* foi desenvolvido em 2002 no INESC Porto [41], [42], sendo que o professor Vladimiro Miranda, Universidade do Porto, teve a ideia original, juntamente com o seu aluno Nuno Fonseca. O *EPSO* é um algoritmo que tem por base a *PSO*, pois existe troca de informação entre as partículas e

um método de computação evolucionária, porque as características (pesos) das soluções são mudadas e transpostas para as gerações futuras, através do operador seleção [41][42]. Ainda assim, os autores optam por considerar o algoritmo sob um ponto de vista totalmente evolucionário. Sob esta perspectiva, deve-se interpretar o algoritmo como um algoritmo autoadaptativo evolucionário, onde o operador recombinação é aperfeiçoado, através do “movimento de partículas” que é proveniente da *PSO*.

O principal objetivo dos autores ao desenvolverem o algoritmo *EPSO* era fornecerem uma nova meta-heurística que fosse aplicada ao campo dos sistemas de energia, apesar da versatilidade do algoritmo tornar possível a aplicação do algoritmo em qualquer outra área científica.

3.4.2 Descrição do algoritmo

O esquema do *EPSO* [42] irá ser estruturado identicamente ao esquema da *PSO*. Cada partícula (solução do problema) é definida pela sua posição (x_i^k , posição da partícula i na iteração k) e velocidade (v_i^k , velocidade da partícula i na iteração k). Como na *PSO*, cada partícula tem acesso à informação da melhor solução encontrada por si ($x_{i,mem}^k$, posição da melhor solução encontrada pela partícula i até à iteração k) e à informação da melhor solução encontrada por toda a população (x_{best} , posição da partícula $best$). Posto isto, cada partícula, representada também por parâmetros estratégicos w , com o decorrer das iterações, irá evoluir seguindo os seguintes passos:

- Replicação: cada partícula é replicada r vezes, resultando num total de $r + 1$ partículas no espaço de pesquisa. Na versão original do *EPSO* foi sempre adotado $r=1$.
- Mutação: cada réplica sofre mutação nos seus parâmetros estratégicos w , da seguinte forma,

$$w_{i,j}^{k+1} = w_{i,j}^k (1 + \tau N(0,1)) \quad 3.6$$

onde τ é o parâmetro de aprendizagem, fixado externamente, e $N(0,1)$ é um número aleatório com distribuição Gaussiana de média 0 e variância 1. O índice j refere-se ao peso de inércia, memória, cooperação e desvio do ótimo. No início do algoritmo, estes parâmetros estratégicos tomam valores aleatórios uniformes entre 0 e 1.

- Reprodução: cada partícula e suas réplicas gera um descendente, de acordo com a equação de movimento, semelhante à equação da *PSO*,

$$x_{best}^{novo} = x_{best} + w_{i,desvOptimo}^k \cdot N(0,1) \quad 3.7$$

$$v_i^{k+1} = w_{i,inercia}^k \cdot v_i^k + w_{i,mem}^k \cdot (x_i^{k,mem} - x_i^k) + w_{i,coop}^k \cdot (x_{best}^{novo} - x_i^k) \quad 3.8$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad 3.9$$

É aplicada uma mutação, expressão 3.7, na melhor solução encontrada pela população. Com este procedimento, o processo autoadaptativo pode focar mais ou menos a orientação

da população e permitir que esta continue a se movimentar, mesmo quando as partículas já convergiram todas para a mesma região do espaço e estão todas muito juntas.

- Avaliação: cada descendente tem a sua aptidão avaliada, de acordo com a sua posição no espaço de busca.
- Seleção: uma partícula, de entre o grupo de $r + 1$ descendentes de cada indivíduo da geração anterior, é seleccionada por torneio estocástico ou outro tipo de seleção.

O esquema do algoritmo *PSO* pode ser ilustrado pela seguinte figura.

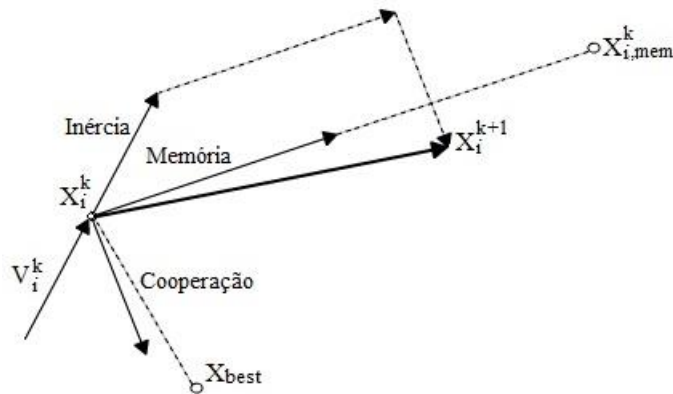


Figura 3.10: Ilustração das três componentes do movimento de uma partícula i . Relativamente ao termo de cooperação, a atração efectua-se para uma vizinhança.

Pode definir-se, então, o algoritmo *EPSO* como um processo $px\sigma AA(1,r+1)$ -ES, o que significa que cada partícula irá criar $r + 1$ descendentes de onde apenas um irá sobreviver. Este processo é realizado para todas as p partículas, sendo que estas estão ligadas através do terceiro termo da equação de movimento, expressão 3.8, o termo de cooperação.

3.5 Partícula

3.5.1 Definição de partícula

A partícula é a essência tanto do algoritmo *PSO* como do algoritmo *EPSO*. O espaço de busca será dividido em D dimensões, como foi descrito na Secção 3.3- Otimização por Enxame de Partículas. No problema OCP, as D dimensões corresponderão a todos os barramentos da rede elétrica que se pretende otimizar. Todas as partículas são consideradas soluções viáveis para a resolução do problema, desde que respeitem as restrições da rede. Como foi referido em “Objetivo da *PSO*”, na secção 3.3- Otimização por Enxame de Partículas, as partículas irão caminhar para uma solução sub ótima ao longo da sua evolução, a partir da informação recolhida da sua experiência e da informação disponibilizada por todas as outras partículas da população.

No problema da OCP, as D dimensões do espaço de busca tomarão valores da capacidade dos bancos de condensadores a implementar em cada barramento da rede. Caso a rede a analisar tenha, por exemplo, quatro barramentos e a capacidade dos bancos de condensadores a implementar permitida tivesse de ser superior a 0 MVar e inferior a 50 MVar, a partícula poderia ser caracterizada como é apresentado na Figura 3.11.

6 Mvar	3 Mvar	22 Mvar	33 Mvar
Bus 1	Bus 2	Bus 3	Bus 4

Figura 3.11: Caracterização simples de uma partícula.

A caracterização da partícula irá depender da informação que esta transportará ao longo da sua evolução. No problema da OCP, a informação essencial e mais relevante será a melhor combinação que essa partícula encontrou até ao momento (b_i), as perdas totais da rede que essa melhor combinação produz e as velocidades da partícula para cada dimensão no espaço de busca. Considerando os mesmos critérios da partícula caracterizada na Figura 3.11, e assumindo que a velocidade da partícula em qualquer dimensão do espaço varia entre -5 e 5, teríamos uma partícula com a caracterização da Figura 3.12.

Posição atual				Melhor posição encontrada pela partícula				Velocidade				Perdas
6 Mvar	3 Mvar	22 Mvar	33 Mvar	5 Mvar	4 Mvar	18 Mvar	40 Mvar	2 Mvar	1 Mvar	3 Mvar	-2 Mvar	6,35 MW
Bus 1	Bus 2	Bus 3	Bus 4	Bus 1	Bus 2	Bus 3	Bus 4	Vel. 1	Vel. 2	Vel. 3	Vel. 4	

Figura 3.12: Caracterização detalhada de uma partícula.

Os primeiros quatro elementos da estrutura da partícula dizem respeito à posição atual da partícula; os quatro elementos seguintes referem-se à melhor combinação de posições encontrada pela partícula (b_i) até ao momento; os últimos cinco elementos são relativos às velocidades da partícula nas quatro dimensões do espaço de busca e às perdas totais da rede com a configuração da melhor combinação de posições encontrada pela partícula. Caso a configuração da rede com as posições atuais da partícula resulte em perdas totais da rede menores que as perdas totais da rede com a configuração b_i , b_i será atualizado, ou seja, os elementos do quinto ao oitavo tomarão os valores dos primeiros quatro elementos. O último elemento também será atualizado, tomando assim o valor das perdas totais da rede com a configuração da combinação das posições atuais da partícula.

A evolução das partículas irá traduzir-se na convergência destas para uma solução sub ótima. Esta evolução irá acontecer devido à atualização da posição e da velocidade da partícula, ou seja, com o avançar das gerações das partículas, as posições das partículas tendem a ficar cada vez melhores, de

acordo com a função objetivo. Os gráficos de convergência das partículas irão depender do objetivo de otimização. Como o pretendido da função objetivo é minimizar as perdas numa rede elétrica, o gráfico de convergência terá uma forma idêntica ao gráfico da Figura 3.13.

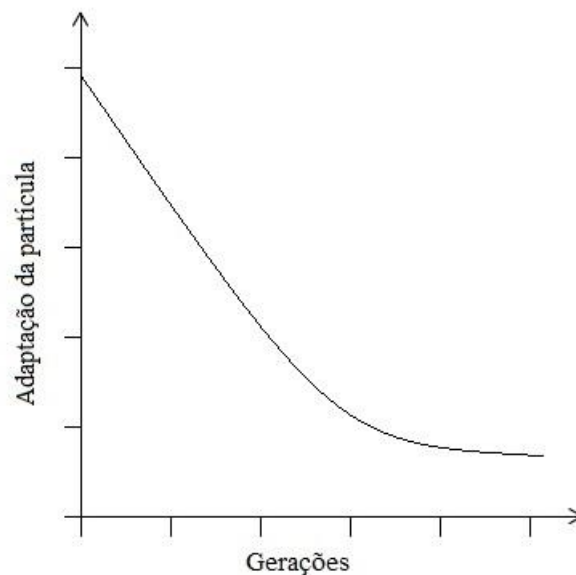


Figura 3.13: Gráficos de convergência das partículas.

3.5.2 Variáveis de decisão e parâmetros estratégicos da partícula

As variáveis de decisão e parâmetros estratégicos serão responsáveis pela caracterização e desempenho da partícula. Serão estes parâmetros que irão definir a partícula e a sua evolução ao longo das gerações, mais concretamente, as variáveis de decisão são responsáveis por caracterizar a partícula e os parâmetros estratégicos serão mais um auxílio para a partícula convergir para uma solução ótima.

Como foi referido em “Natureza do problema”, na secção 2.1.2, as variáveis de decisão no problema *OCP* serão a capacidade dos bancos de condensadores a implementar em cada barramento da rede elétrica, ou seja, caso a rede tenha quatro barramentos, a partícula irá ser composta por quatro variáveis de decisão. As variáveis de decisão têm de estar compreendidas entre um intervalo de valores, no problema da *OCP* o intervalo será entre a capacidade mínima e máxima do banco de condensadores. O valor mínimo das variáveis de decisão poderá ser estipulado como negativo e, nesse caso, quando uma variável de decisão tomar um valor negativo, deixamos de estar perante a implementação de reatâncias capacitivas e passamos a estar perante a implementação de reatâncias indutivas. Para garantir que nenhum barramento esteja sobrecompensado, nos barramentos em que existam cargas, o módulo da capacidade do banco de condensadores a implementar nesses barramentos deve ser sempre inferior ao módulo da potência reativa da carga, expressão 2.5. Quando o algoritmo é iniciado, seja o algoritmo da *PSO* ou da *EPSO*, tanto as variáveis de decisão como as velocidades da partícula são aleatoriamente iniciadas dentro dos valores limites aceitáveis.

Relativamente aos parâmetros estratégicos já existirão algumas diferenças do algoritmo da *PSO* face ao algoritmo da *EPSO*. No algoritmo da *PSO* existirá apenas uma variável estratégica, o peso da inércia, w . A seleção adequada do peso da inércia é traduzida num equilíbrio entre a exploração global e local do espaço de busca. Embora estes algoritmos sejam desenvolvidos para resolverem problemas de soluções estáticas, os problemas do mundo real mudam de estado frequentemente (ou continuamente). Apesar deste facto, no algoritmo da *PSO*, o peso da inércia não será do tipo variável mas sim do tipo constante, inicializada quando o algoritmo da *PSO* é iniciado. Como foi referido em “Descrição do algoritmo”, na secção 3.3, o peso da inércia foi primeiramente introduzido por Eberhart e Shi[37], em que os autores aconselhavam a estabelecer o valor de w entre o intervalo de $[0.8,1.2]$. Posteriormente, Clerc[43] introduziu o método do fator de contração, onde ϕ é estipulado a 4.1e a constante K igual a 0.729, como se pode observar pelas expressões 3.10 e 3.11.

$$V_i(t) = K \cdot [V_i(t-1) + c_1 \cdot rand_1 \cdot (b_i - X_i) + c_2 \cdot rand_2 \cdot (b_G - X_i)] \quad 3.10$$

$$K = \frac{2}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|}, \text{ onde } \phi = c_1 + c_2, \phi > 4 \quad 3.11$$

A partir do método de Clerc e usando a equação de velocidade, expressão 3.5, o peso da inércia, w , é estabelecido a 0.7 e as constantes c_1 e c_2 igualadas a 2.

Como foi exposto em “Descrição do algoritmo”, na secção 3.4, a equação da velocidade da *EPSO* difere da equação da velocidade da *PSO*. A diferença incide no facto de cada componente da equação da velocidade ter um parâmetro estratégico denominado peso, sendo eles o peso da inércia, da memória e da cooperação. A melhor combinação encontrada até determinada geração (b_G) sofrerá mutações que introduzem também outro parâmetro estratégico, o peso do desvio ótimo. A mutação destes quatro parâmetros estratégicos introduz o parâmetro estratégico τ , denominado de parâmetro de aprendizagem. O parâmetro de aprendizagem deve ser suficientemente pequeno, τ igual a 0.1, para garantir o controlo da mutação de modo a não aparecerem pesos negativos.

Outro parâmetro estratégico do algoritmo da *EPSO* é o fator de comunicação, p . Este parâmetro representa um papel muito importante na fase final de convergência. Como o próprio nome indica, este parâmetro é responsável pela comunicação, ou por outras palavras, troca de informação entre partículas. Nas primeiras gerações, a partilha da informação da melhor posição (b_G) encontrada pelas partículas da população é essencial para que as partículas convirjam para o local ótimo do espaço de busca; mas quando as partículas já se encontram, praticamente todas, muito próximas do local ótimo, a informação de b_G faz com que as partículas não explorem a vizinhança próxima de b_G . Posto isto, é introduzido o fator de comunicação que é responsável por existir ou não a componente cooperação na equação da velocidade. Caso o número *rand* for inferior ou igual ao fator de comunicação, a componente de cooperação integra na equação da velocidade, caso contrário, não integra, como é

exemplificado pelo sistema de equações 3.12. O valor do fator de comunicação foi estabelecido em 0.2.

$$v_i^{k+1} = \begin{cases} w_{i,inercia}^k \cdot v_i^k + w_{i,mem}^k \cdot (x_i^{k,mem} - x_i^k) + w_{i,coop}^k \cdot (x_{best}^{novo} - x_i^k), & \text{rand} \leq p \\ w_{i,inercia}^k \cdot v_i^k + w_{i,mem}^k \cdot (x_i^{k,mem} - x_i^k), & \text{rand} > p \end{cases} \quad 3.12$$

3.5.3 Avaliação da partícula

A avaliação da partícula consiste numa função que irá avaliar a aptidão da partícula, ou seja, uma função que irá retornar o valor *fitness* da partícula. Como foi referido em “Formulação do problema”, na secção 2.2, o problema consiste na redução de perdas numa rede elétrica com visa a otimizar essa rede. Assim, o valor *fitness* da partícula será as perdas totais da rede com implementação dos bancos de condensadores referentes à combinação de posições da partícula. Visto que se pretende minimizar as perdas, quanto menor for o valor *fitness* melhor será a aptidão da partícula. O valor *fitness* de uma partícula não irá estar inteiramente ligado apenas às perdas totais da rede devido às restrições da rede. Estas restrições dizem respeito a tensões de barramento fora dos estipulados, expressão 3.13, e a sobrecompensações. A restrição da sobrecompensação nunca acontece, pois o espaço de busca é limitado nos barramentos com cargas. Como se pode observar na 3.13, os limites para as tensões de barramento foram estipulados 0.95 para V_{min} e 1.05 para V_{max} .

$$0.95 \leq V_{pu} \leq 1.05 \quad 3.13$$

Posto isto, penalidades foram implementas no valor *fitness* de uma partícula. Caso a tensão num barramento não esteja dentro dos limites, o valor *fitness* da partícula irá ser 100. Este valor excessivo transmite a ideia de que uma partícula, que faça com que a rede apresente tensões de barramento fora dos limites estipulados, não seja de nenhuma maneira uma solução para o problema da OCP. Deste modo, o valor *fitness* pode ser descrito pela sistema de equações 3.14.

$$fitness = \begin{cases} \sum_{i=0}^{n-1} P_{loss(i,i+1)}, & Vb = 0 \\ 100, & Vb > 0 \end{cases} \quad 3.14$$

Onde n é o o número total de barramentos da rede elétrica em estudo e Vb o número total de barramentos com tensão fora dos limites estipulados. O valor da penalidade da tensão deve ser adequado à rede em estudo. Nas redes analisadas, o valor escolhido cumpriu com o objetivo da penalidade.

4. Aplicação dos algoritmos

Aplicação dos algoritmos

às redes elétricas em estudos.

Apresentação e discussão dos resultados obtidos.

4.1 Arquitetura do programa

Até este capítulo apenas se falou do algoritmo capaz de resolver o problema *OCP*. O algoritmo é a ferramenta mais importante para a resolução deste problema, mas outras ferramentas têm de ser manipuladas para se obter os resultados desejados.

Cada vez que uma partícula se desloca no espaço de busca, uma nova solução é proposta, ou seja, uma nova rede tem de ser analisada. A análise dessa rede é feita através do programa *PSS/E*, versão 33. O programa *PSS/E* retorna as perdas elétricas totais da rede e as tensões nos barramentos.

O algoritmo é controlado por uma aplicação responsável por gerar, avaliar e selecionar partículas do enxame. Quando uma partícula se desloca no espaço de busca, esta aplicação envia a nova configuração da rede correspondente às novas posições partícula para o programa *PSS/E*, manda correr o fluxo de potência (*Power Flow*) e recebe os resultados. Com estes resultados, a aplicação é capaz de avaliar a partícula e selecioná-la como melhor solução encontrada até ao momento, caso a aptidão da partícula seja a melhor do enxame.

A linguagem escolhida para programar esta aplicação foi *Python* [44], versão 27. A escolha recaiu nesta linguagem devido à sua fácil adaptação e à API (*Application Program Interface*) que o programa *PSS/E* disponibiliza relativamente a esta linguagem, denominado módulo *PSSPY*.

O programa que foi utilizado para desenvolver a aplicação foi o *Eclipse* [45]. Um ambiente de desenvolvimento integrado com *debug* em tempo real.

A base de dados é também monitorizada pela aplicação que controla o algoritmo. O ficheiro que contém os resultados obtidos é produzido pelo programa *Excel*. Este programa também tem uma API dedicada à linguagem *Python*, o módulo *EXCELPY*, que facilita o armazenamento da informação importante. Esta base de dados serve apenas para armazenar dados, ou seja, a aplicação não é capaz de usar dados da base de dados apenas envia os dados para armazenar.

Na Figura 4.1 é ilustrado o esquema da arquitetura do programa onde se observa a arquitetura da aplicação.

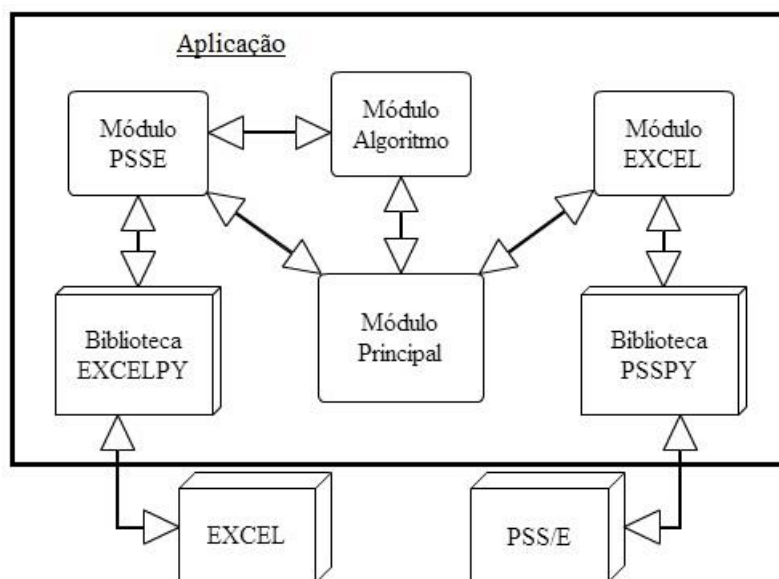


Figura 4.1: Arquitetura do programa englobando a arquitetura da aplicação.

Os módulos que constituem a aplicação são:

- Módulo Principal: é o módulo responsável por coordenar e transmitir a informação entre os restantes módulos.
- Módulo Algoritmo: é o módulo onde está implementado o algoritmo *PSO* e *EPSO*. O código pode ser observado no anexo VI.
- Módulo EXCEL: através das funções da biblioteca EXCELPY é criado o ficheiro onde irão ser armazenados os dados mais importantes.
- Módulo PSS/E: através das funções da biblioteca PSSPY é enviado para o programa PSS/E os dados relevantes para modificar a rede e recebidos os resultados do *power flow* da rede em análise.

Exteriores à aplicação existem os módulos:

- PSS/E: é o programa responsável por correr o *power flow* da rede em análise.
- Excel: é o programa que gera a base de dados.

4.2 Algoritmos aplicados

4.2.1 Aplicação do algoritmo *PSO* ao problema *OCP*

O algoritmo da *PSO* pode ser aplicado ao problema *OCP* por dois métodos diferentes. Estes métodos diferem no momento de atualização da melhor posição encontrada pelo conjunto total de partículas (b_G) que irá ser fundamental para a equação da velocidade, expressão 3.4.

No primeiro método, a atualização será feita no final da geração, ou seja, depois de todas as partículas terem sofrido uma translação no espaço através da equação da velocidade, onde o valor de b_G será o melhor encontrado até à geração anterior.

Enquanto, no segundo método, a atualização será feita assim que cada partícula sofra uma translação no espaço através da equação da velocidade. Caso uma partícula, depois de sofrer a translação no espaço, encontre uma solução melhor que a atual b_G , b_G será atualizado e usado na equação da velocidade para a transladação da próxima partícula. De seguida são apresentados os fluxogramas que exemplificam a diferença destes dois métodos, sendo que a Figura 4.2(a) é referente ao método de atualização no final da geração e a Figura 4.2(b) ao outro método.

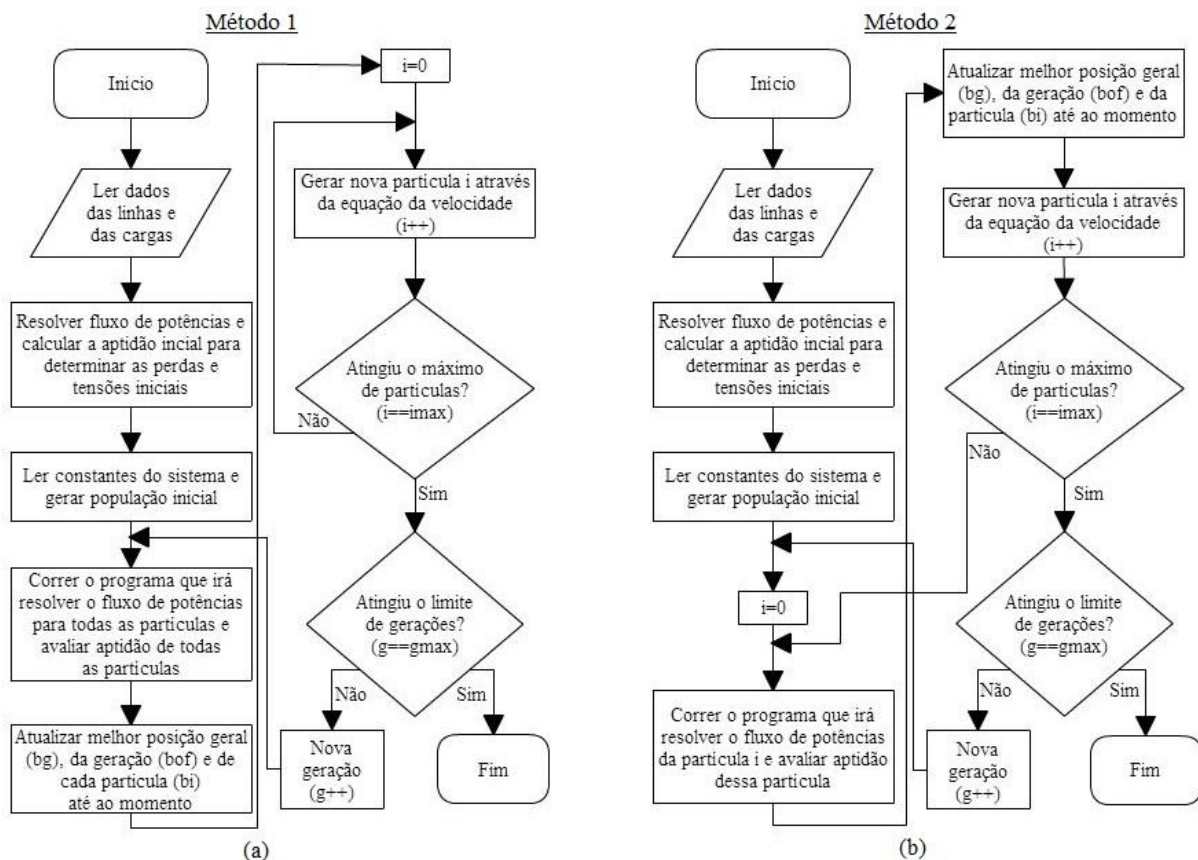


Figura 4.2: Fluxogramas dos procedimentos principais das duas abordagens da *PSO*.

4.2.2 Aplicação do algoritmo *EPSO* ao problema *OCP*

Da mesma maneira como foi aplicado o algoritmo da *PSO* ao problema *OCP*, será aplicado o algoritmo da *EPSO*. Desta forma, os fluxogramas da Figura 4.3 exemplificam os dois métodos idênticos aos métodos da *PSO*, sendo que a Figura 4.3(a) diz respeito ao método de atualização da melhor solução encontrada no final de cada geração e a Figura 4.3(b) é referente ao método de atualização da melhor solução encontrada sempre que uma partícula sofra uma translação no espaço de busca.

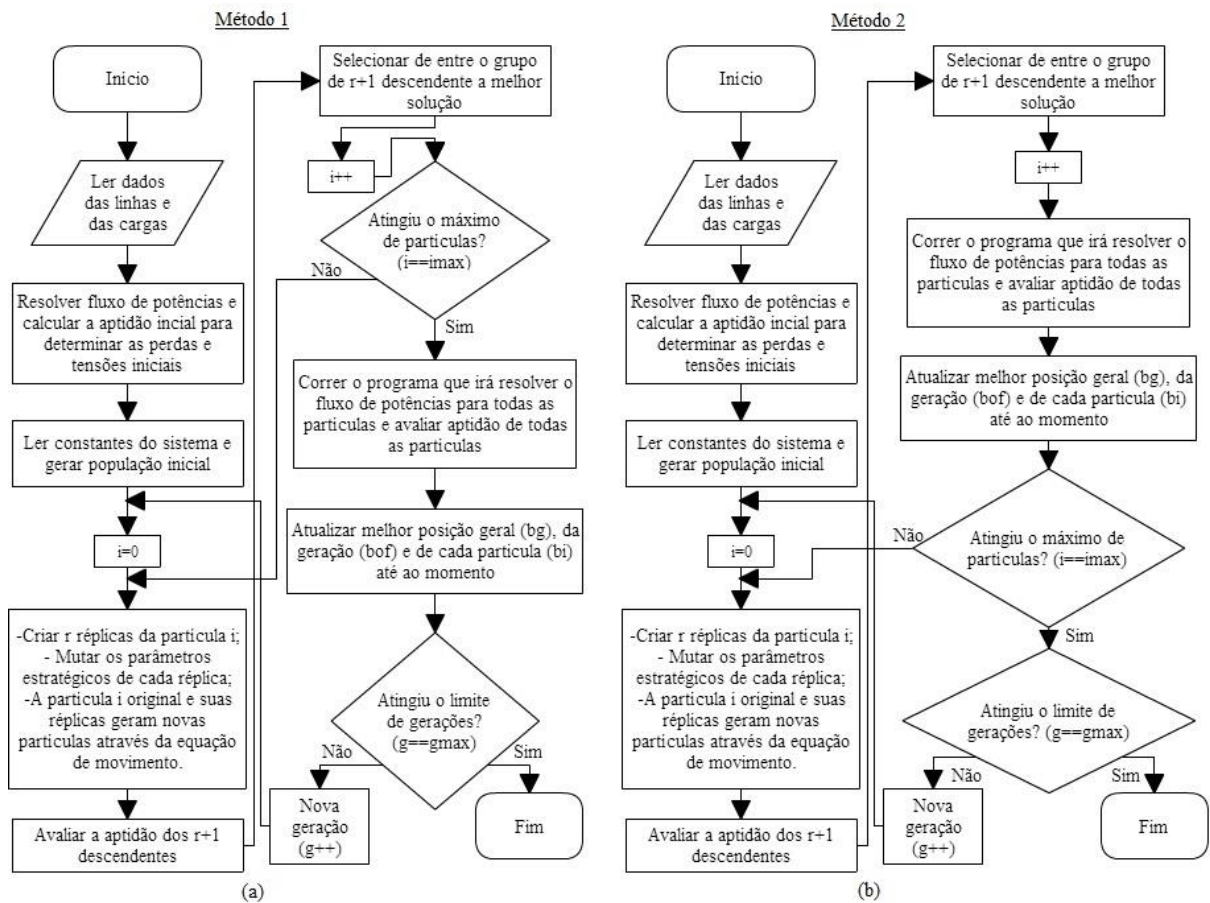


Figura 4.3: Fluxogramas dos procedimentos principais dos dois métodos da *EPSO*.

Depois de efetuados testes usando ambos os métodos foi escolhido o método 1 devido à melhor qualidade dos resultados apresentados. O método escolhido foi aplicado a ambos os algoritmos, *PSO* e *EPSO*, assim a atualização da melhor solução encontrada é feita sempre no final de cada geração. No Anexo VI, está presente o código de ambos os algoritmos e pode observar-se tanto no ficheiro *pso_swarm* e *epso_swarm* que a atualização é feita no fim da geração.

4.3 Parametrização dos algoritmos

Como tem sido referido, os algoritmos utilizados para a resolução do problema *OCP* foram o *PSO* e o *EPSO*. Estes algoritmos têm vários parâmetros com grande influência nos resultados obtidos, isto é, uma escolha acertada no conjunto de parâmetros irá equivaler a resultados consistentes, ao passo que se os parâmetros escolhidos não forem os adequados para a rede elétrica em estudo os resultados não irão ser os pretendidos.

Os parâmetros dizem respeito ao número de partículas, posições máximas e mínimas das partículas no espaço de busca, velocidades máximas e mínimas da partícula, número de gerações máximas de uma partícula e os valores das penalidades.

O utilizador, ao analisar uma rede elétrica, deve, à partida, efetuar alguns testes para experimentar valores para estes parâmetros. Só através da experiência se conseguirá obter bons resultados. De seguida, demonstrar-se-á como os parâmetros influenciam os resultados do algoritmo.

Inicialmente são escolhidos determinados parâmetros. Através da análise dos resultados obtidos, constata-se quais os parâmetros que estão ajustados incorretamente à rede em estudo, procede-se, então, à alteração dos mesmos para que os resultados se traduzam numa solução sub ótima para o problema. Serão feitos dez testes usando o algoritmo *PSO*. Como exemplo, considera-se a rede em estudo de seis barramentos [46], Figura 4.4, para um único cenário de carga. Os dados da rede podem ser consultados no Anexo II.

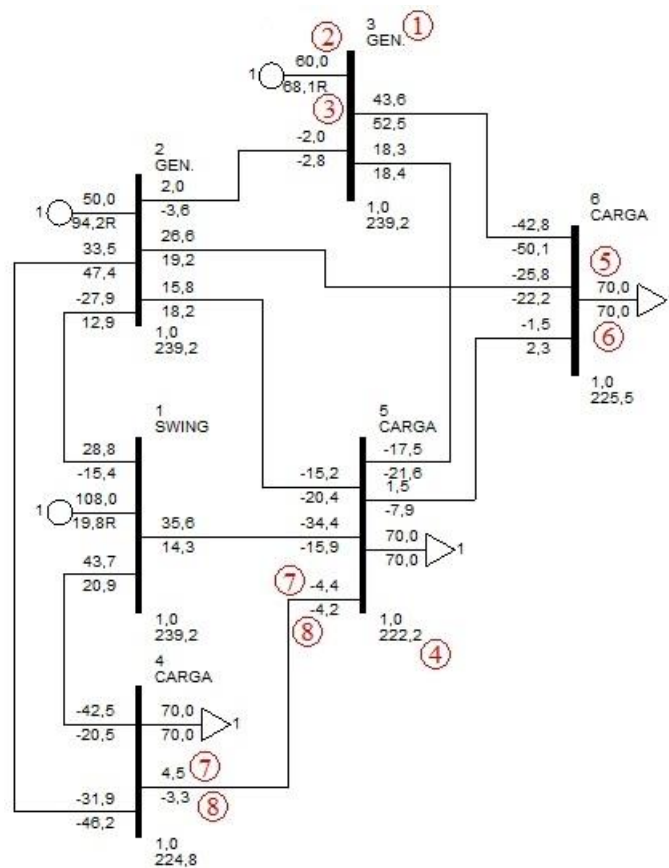


Figura 4.4: Esquema da rede original de seis barramentos.

Legenda:

1. Nome e número do barramento.
2. Potência ativa gerada.
3. Potência reativa gerada.
4. Tensão no barramento, expressa em pu e kV.
5. Potência ativa da carga.
6. Potência reativa da carga.
7. Potência ativa que entra (-) ou sai (+) do barramento.
8. Potência reativa que entra (-) ou sai (+) do barramento.

4.3.1 PSO

Na primeira fase de testes, foram estipulados os parâmetros apresentados na Tabela 3. Na Tabela 4 podem observar-se os resultados obtidos com estes parâmetros. É claro que a capacidade máxima do banco de condensadores a instalar num barramento é reduzida, pois em todos os testes o algoritmo apresenta como melhor solução uma configuração da rede, onde nos três últimos barramentos a capacidade do banco de condensadores é a máxima permitida.

Tabela 3: Parâmetros escolhidos para a primeira fase de testes.

Partículas	Gerações	Capacidade Máxima	Capacidade Mínima	Velocidade Máxima	Velocidade Mínima
50	100	50	-50	10	-10

Tabela 4: Resultados obtidos na primeira fase de testes utilizando o algoritmo *PSO*.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
		1	2	3	4	5	6	
1	4.474	0	0	0	50	50	50	150
2	4.474	0	0	0	50	50	50	150
3	4.474	0	0	0	50	50	50	150
4	4.474	0	0	0	50	50	50	150
5	4.474	0	0	0	50	50	50	150
6	4.474	0	0	0	50	50	50	150
7	4.474	0	0	0	50	50	50	150
8	4.474	0	0	0	50	50	50	150
9	4.474	0	0	0	50	50	50	150
10	4.474	0	0	0	50	50	50	150

Na segunda fase de testes, a capacidade máxima foi alterada para 100 MVar, como se pode ver na Tabela 5. Como se pode constatar na Tabela 6, o algoritmo apresenta a mesma solução para todos os testes, sendo esta a solução ideal para a rede em estudo. Analisando os resultados apresentados, também se percebe que o parâmetro capacidade mínima pode ser menor, apesar de ser irrelevante para a convergência do algoritmo.

Tabela 5: Parâmetros escolhidos para a segunda fase de testes.

Partículas	Gerações	Capacidade Máxima	Capacidade Mínima	Velocidade Máxima	Velocidade Mínima
50	100	100	-50	10	-10

Tabela 6: Resultados obtidos na segunda fase de testes utilizando o algoritmo *PSO*.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
		1	2	3	4	5	6	
1	4,311	0	0	0	66	58	62	186
2	4,311	0	0	0	66	58	62	186
3	4,311	0	0	0	66	58	62	186
4	4,311	0	0	0	66	58	62	186
5	4,311	0	0	0	66	58	62	186
6	4,311	0	0	0	66	58	62	186
7	4,311	0	0	0	66	58	62	186
8	4,311	0	0	0	66	58	62	186
9	4,311	0	0	0	66	58	62	186
10	4,311	0	0	0	66	58	62	186

Analisando o gráfico de convergência, Figura 4.5, onde se pode verificar a melhor solução encontrada em cada geração e a melhor solução global, de um teste aleatório de entre os dez feitos, verifica-se que, a partir da décima geração, o algoritmo já encontrou a solução que apresenta as perdas totais mínimas da rede em estudo. Sendo assim, em termos de eficiência, o parâmetro gerações ou até o número de partículas da população pode ser reduzido para que o algoritmo apresente menos tempo de computação tornando-o mais eficiente.

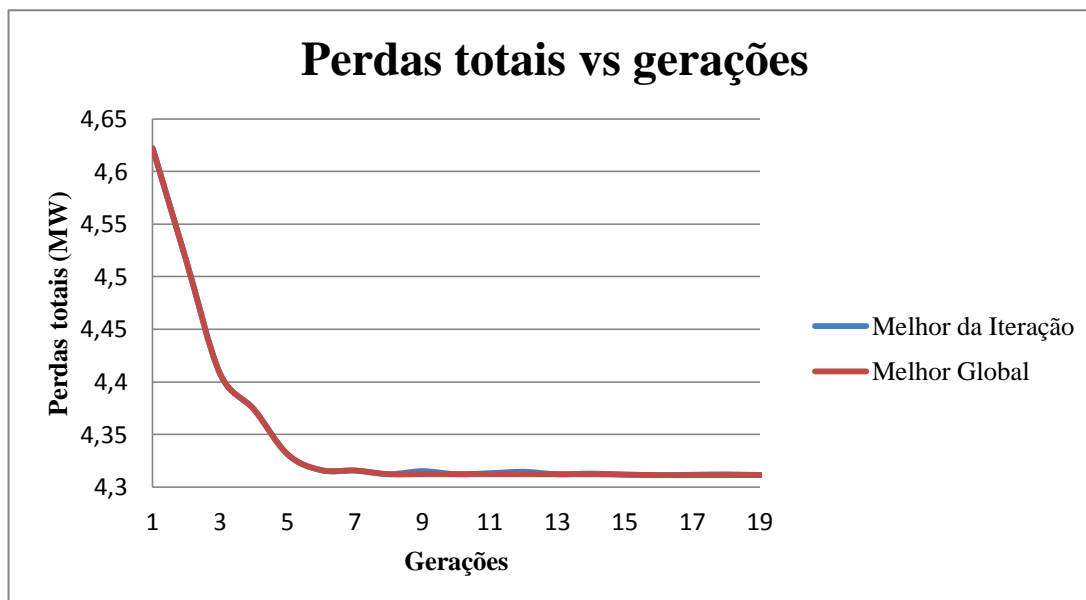


Figura 4.5: Gráfico de convergência de um teste da segunda fase de testes.

Os parâmetros número de gerações e número de partículas foram então definidos, como se pode observar na Tabela 7. Os resultados obtidos são apresentados na Tabela 8, e, como esperado, pode constatar-se que o algoritmo apresenta a solução ótima em todos os testes. Em termos de eficiência, um teste com os parâmetros da Tabela 5 demora cerca de 44 segundos, ao passo que um teste com os parâmetros da Tabela 7 demora cerca de 22 segundos.

Tabela 7: Parâmetros escolhidos para a terceira fase de testes.

Partículas	Gerações	Capacidade	Capacidade	Velocidade	Velocidade
		Máxima	Mínima	Máxima	Mínima
30	80	100	-10	10	-10

É notório que a escolha dos parâmetros do algoritmo irá ser determinante nos resultados obtidos. Ou seja, se a escolha dos parâmetros for ajustada ao perfil da rede elétrica, os resultados obtidos serão sempre próximos da solução ótima. Visto que o principal objetivo é encontrar uma solução que se traduza em perdas mínimas totais da rede, os parâmetros fulcrais serão a capacidade máxima e mínima, isto é, a posição máxima e mínima da partícula no espaço de busca. Os parâmetros velocidade máxima e mínimos também são importantes, pois simbolizam que distância pode andar uma partícula no espaço de busca de geração em geração. Embora nesta rede de seis barramentos a velocidade não seja tão influente como em outras redes onde exista mais barramentos, o que reforça a ideia de que os parâmetros têm determinada importância, consoante a rede em estudo.

Tabela 8: Resultados obtidos na terceira fase de testes utilizando o algoritmo *PSO*.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
		1	2	3	4	5	6	
1	4,311	0	0	0	66	58	62	186
2	4,311	0	0	0	66	58	62	186
3	4,311	0	0	0	66	58	62	186
4	4,311	0	0	0	66	58	62	186
5	4,311	0	0	0	66	58	62	186
6	4,311	0	0	0	66	58	62	186
7	4,311	0	0	0	66	58	62	186
8	4,311	0	0	0	66	58	62	186
9	4,311	0	0	0	66	58	62	186
10	4,311	0	0	0	66	58	62	186

Este facto pode ser verificado na Figura 4.6, onde é explícito que, em ambos os testes, o algoritmo converge para a solução ótima sendo que, como era esperado, quando o parâmetro da velocidade é maior, a convergência para a solução ideal acontece com menor número de gerações. Os parâmetros que estão ligados com a eficiência do algoritmo podem ser postos em segundo plano, pois o principal objetivo do algoritmo é resolver o problema *OCP*, logo é preferível usar parâmetros que garantam seguramente a convergência do algoritmo para a solução sub ótima do que parâmetros que, apesar de acelerarem o processo do algoritmo, possam apresentar resultados não tão satisfatórios. Mais uma vez é de acentuar que a escolha dos parâmetros tem de ser concordante com a topologia da rede elétrica em estudo. Por exemplo, se a rede apresentar perfis de carga elevada, a capacidade máxima dos condensadores deve ser elevada; se a rede for composta por muitos barramentos, as combinações de soluções possíveis serão inúmeras, sendo assim preferível aumentar o número de gerações ou o número de partículas para garantir a solução ótima.

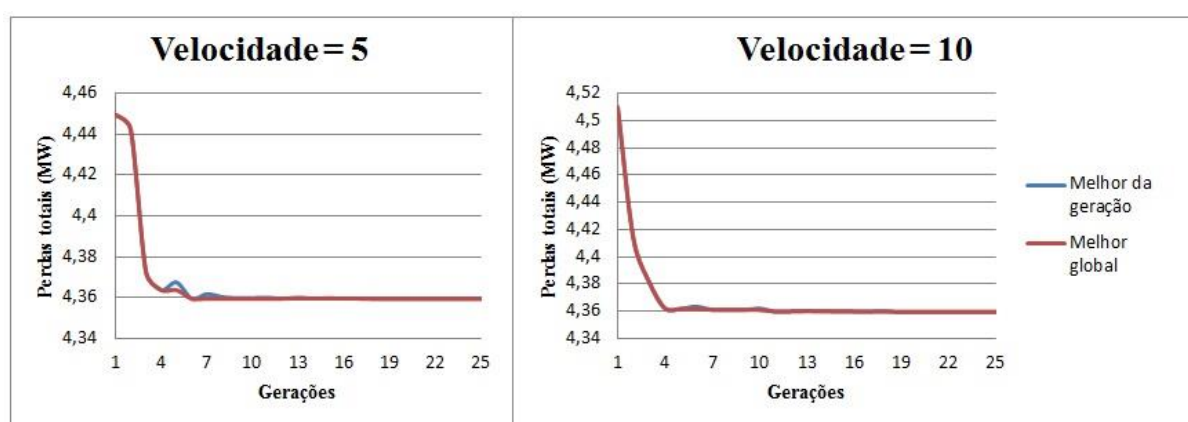


Figura 4.6: Gráficos de convergência para diferentes velocidades.

4.3.2 EPSO

Aplicando o algoritmo *EPSO* nas mesmas condições (parâmetros da Tabela 7 e a rede de seis barramentos) que foram feitos os testes para o algoritmo *PSO*, obtemos os resultados expressos na Tabela 9. Como se pode observar, os parâmetros que funcionam na perfeição para o algoritmo *PSO*, para o *EPSO* verifica-se que não são os ideais. Apesar de todas as melhores soluções encontradas pelo algoritmo nos dez testes apresentarem sempre as mínimas perdas totais possíveis para a rede em estudo, a capacidade total dos bancos de condensadores apenas em cerca de metade dos testes se aproxima da mínima.

Tabela 9: Resultados obtidos na primeira fase utilizando o algoritmo *EPSO*.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
		1	2	3	4	5	6	
1	4,311	10	0	2	66	58	62	198
2	4,311	0	0	17	66	58	62	203
3	4,311	4	0	0	67	58	62	191
4	4,311	0	7	0	66	58	62	193
5	4,311	0	0	0	66	58	62	186
6	4,311	0	0	6	67	58	62	193
7	4,311	0	0	3	66	58	62	189
8	4,311	0	1	0	66	58	62	187
9	4,311	0	0	43	66	58	62	229
10	4,311	0	0	0	66	58	62	186

Analisando os resultados, é perceptível que o algoritmo *EPSO* necessita de mais gerações para facilitar a convergência para a solução ideal. Sendo assim, ao invés de haver trinta partículas e oitenta gerações, passamos a ter quarenta partículas e cem gerações, como se pode observar na Tabela 10. Os resultados obtidos com os novos parâmetros podem ser verificados na Tabela 11.

Tabela 10: Parâmetros escolhidos para a segunda fase utilizando o algoritmo *EPSO*.

Partículas	Gerações	Capacidade Máxima	Capacidade Mínima	Velocidade Máxima	Velocidade Mínima
40	100	100	-10	10	-10

Tabela 11: Resultados obtidos utilizando o algoritmo *EPSO*.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
		1	2	3	4	5	6	
1	4,311	0	0	0	66	58	62	186
2	4,311	0	0	0	66	58	62	186
3	4,311	0	0	2	66	58	62	188
4	4,311	0	0	0	66	58	62	186
5	4,311	0	0	0	66	58	62	186
6	4,311	1	0	1	66	58	62	188
7	4,311	0	0	0	66	58	62	186
8	4,311	0	1	0	66	58	62	187
9	4,311	0	0	0	66	58	62	186
10	4,311	0	0	0	66	58	62	186

Ainda que os resultados obtidos não representem uma unânime solução ótima, nos dez testes, só apenas em três é que a solução não é a ótima. A diferença de capacidade total da solução ótima para as restantes é pequena, variando de um a dois MVar, o que torna estas soluções muito próximas do ótimo. A convergência para a capacidade total mínima possível é alcançada nas últimas gerações para o algoritmo *PSO* e para o algoritmo *EPSO*, como se pode observar no Anexo I.

Na fase final da convergência, as partículas já se encontram todas relativamente perto umas das outras, o que faz com que a componente memória e cooperação da equação movimento não tenham grande influência na velocidade da partícula. A componente inércia, nesta fase, irá ser a responsável por as partículas tentarem encontrar soluções melhores que a melhor global até ao momento. Posto isto, e analisando a Figura 4.7, é explicado o porquê de a convergência demorar mais tempo aplicando o algoritmo *EPSO* face à aplicação do algoritmo *PSO*.

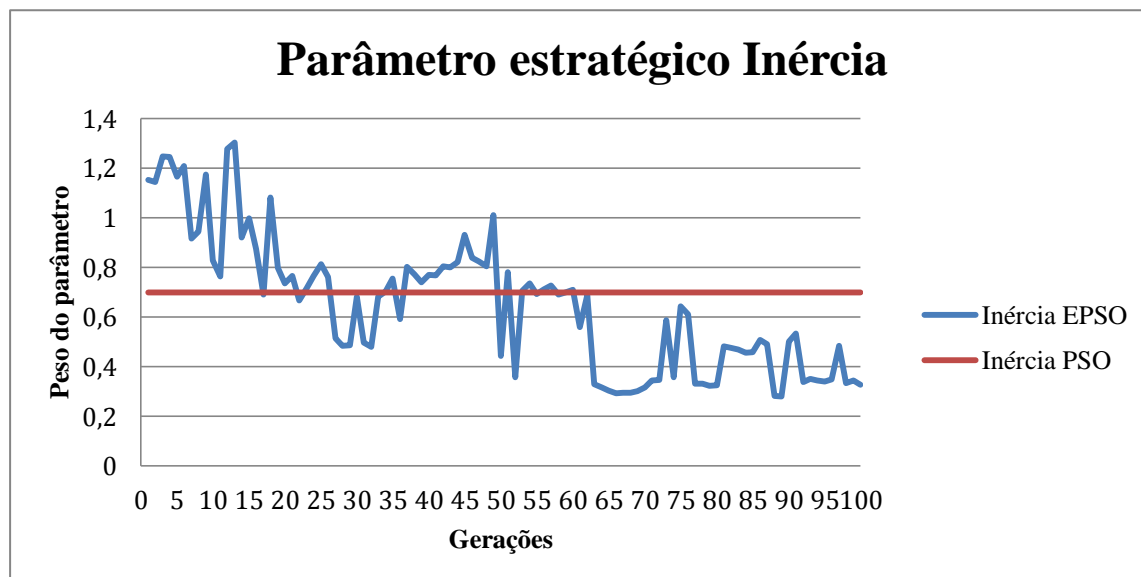


Figura 4.7: Diferença entre o peso do parametro estratégico inércia do algoritmo *PSO* e *EPSO* com o avançar das gerações.

4.4 Aplicação dos algoritmos às redes em estudo

As redes em estudo que irão servir de modelos de teste para os dois algoritmos, *PSO* e *EPSO*, são as redes de seis, catorze, fornecidas pelo *IEEE*, e outra rede de vinte e um barramentos. Estas redes são excelentes para testar algoritmos de otimização de redes elétricas, pois, normalmente, são mal condicionadas, apresentando grandes perdas, linhas em sobrecarga e grandes potências geradas no barramento de balanço.

Como tem sido referido, as n dimensões das partículas correspondem à capacidade do banco de condensadores a instalar nos n barramentos da rede. O espaço de busca da partícula irá estar limitado às capacidades mínimas e máximas admissíveis nodalmente, como foi explicado na Secção 3.5.2. A partícula movimenta-se no espaço de busca a um passo mínimo de 1 e máximo igual a velocidade máxima ou mínima, caso a mínima seja um valor negativo. Caso existam bancos de condensadores na rede em estudo, esses bancos de condensadores ficarão inativos de modo a não interferirem no processo do algoritmo.

4.4.1 Rede “IEEE6”

A rede original de seis barramentos [46], “IEEE6”, por conveniência, é novamente apresentada, Figura 4.8. É ilustrada a percentagem de ocupação das linhas da rede original de seis barramentos e, como se pode observar, existem muitas linhas em sobrecarga ou perto da capacidade máxima. A rede não sofreu qualquer tipo de modificação antes de ser aplicado o algoritmo. No Anexo II são discriminados os dados referentes à rede onde o principal a reter são as perdas iniciais totais da rede, 8.033 MW.

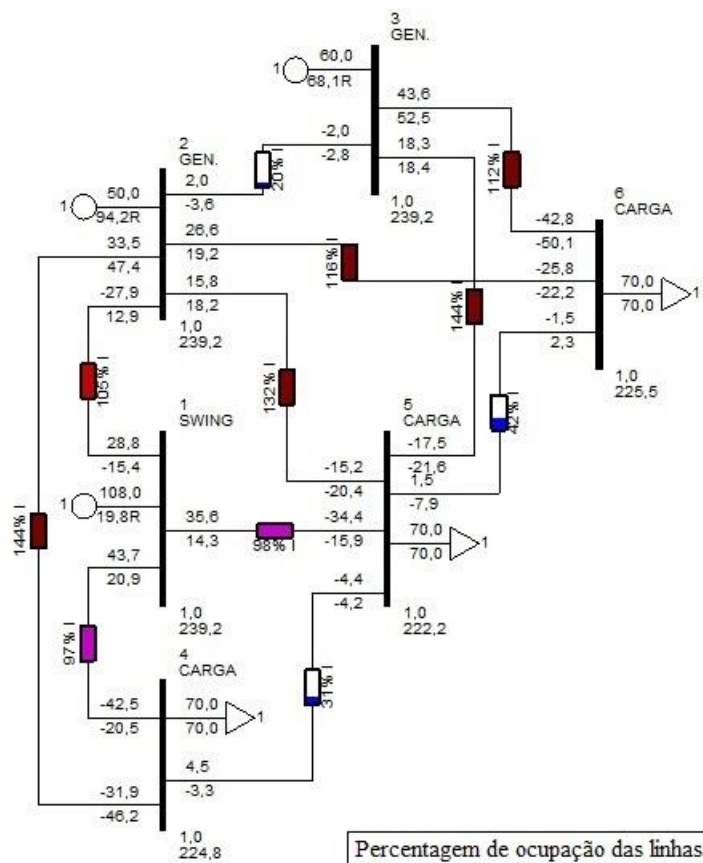


Figura 4.8: Esquema da rede original de seis barramentos.

Aplicação do algoritmo *PSO*

Antes de aplicar o algoritmo *PSO* para resolver o problema *OCP* da rede de seis barramentos, é necessário escolher os parâmetros que foram abordados na seção 4.1. Como a rede de seis barramentos serviu de exemplo para explicar a escolha dos parâmetros, é claro que os parâmetros que irão ser utilizados são os expressos na Tabela 7. A melhor solução encontrada pelo algoritmo *PSO* é apresentada na Tabela 12.

Tabela 12: Melhor solução encontrada pelo algoritmo *PSO*, referente à rede de seis barramentos.

Barramento	1	2	3	4	5	6
Capacidade [MVar]	0	0	0	66	58	62
Perdas totais (MW) – 4.311						

Na Figura 4.9 é apresentada a rede de seis barramentos com a implementação dos bancos de condensadores correspondentes à configuração da melhor solução encontrada pelo algoritmo *PSO*. São também exibidas as percentagens de ocupação das linhas, onde se faz notar a ausência de linhas em sobrecarga.

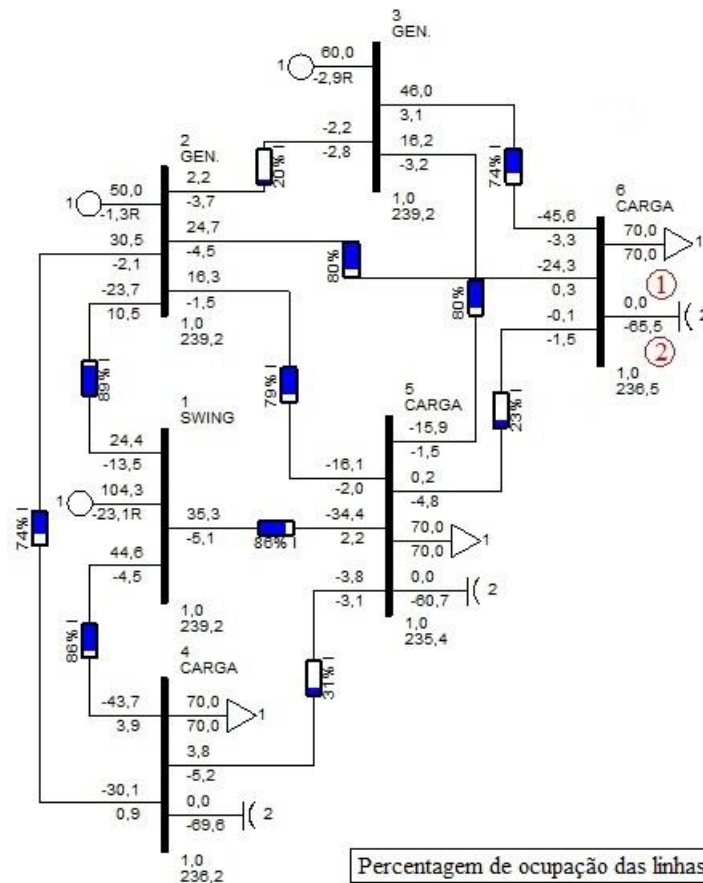


Figura 4.9: Esquema da rede de seis barramentos, depois de aplicado o algoritmo *PSO*.

Legenda:

1. Potência ativa gerada pelo shunt.
2. Potência reativa gerada pelo shunt.

Aplicação do algoritmo *EPSO*

Como foi mencionado no parágrafo anterior, a rede em estudo é a mesma que serviu de exemplo na secção 4.1, logo os parâmetros utilizados na aplicação do algoritmo *EPSO* serão os representados na Tabela 10. A melhor solução encontrada pelo algoritmo *EPSO* foi igual à melhor solução encontrada pelo algoritmo *PSO*, por conseguinte, a configuração da rede elétrica será a mesma que a apresentada na Figura 4.9. As perdas elétricas de todas as linhas e as perdas totais para o caso da rede original de

seis barramentos e para os casos da rede com implementação de bancos de condensadores estão apresentadas no Anexo V.

Análise dos resultados

A melhor solução encontrada foi a mesma em ambos os algoritmos, o que não é estranho. Como a rede é apenas constituída por seis barramentos as combinações de soluções possíveis são reduzidas. Analisando os resultados da melhor solução encontrada, é claro que a redução de perdas totais é significativa. A rede de seis barramentos original apresentava perdas totais iguais a 8.033 MW. A rede de seis barramentos, com a implementação dos bancos de condensadores correspondentes à melhor solução encontrada, apresenta perdas totais iguais a 4.311 MW, o que significa uma redução de cerca de 46%.

A rede de seis barramentos apresenta três barramentos com cargas, sendo que as três cargas são iguais. A localização dos bancos de condensadores era a esperada, pois estes irão compensar a potência reativa de caráter indutivo proveniente das cargas, assim a melhor solução encontrada determinou que devem ser implementados bancos de condensadores nos barramentos 4, 5 e 6. A combinação de capacidades dos bancos de condensadores proposta pela melhor solução encontrada deve-se somente à topologia da rede. Analisando a Figura 4.9, a capacidade dos bancos de condensadores implementados é diferente. Na tabela 1 do Anexo V, observa-se que as linhas que ligam o barramento 4 aos barramentos de geração 1 e 2 são as linhas que apresentam as maiores perdas, assim o banco de condensadores implementado no barramento 4 tem a capacidade suficiente para suprimir a potência reativa da carga, desta forma a potência reativa das linhas reduzirá o que faz com que a corrente que circula nessas linhas também reduza. Como se verifica nessa mesma tabela, as maiores diminuições de perdas foram nas linhas que ligam o barramento 3 aos barramentos 5 e 6 e na linha que liga o barramento 2 ao barramento 4. Esta ocorrência é explicada, pois a potência reativa nestas linhas reduziu, diminuindo desta forma a potência aparente, expressão 4.1, o que se traduz na melhoria do fator de potência, expressão 4.2. As perdas são proporcionais ao quadrado da corrente, expressão 2.1, e como a corrente é reduzida na razão direta da melhoria do fator de potência, expressão 4.3, podemos afirmar que as perdas das linhas são inversamente proporcionais ao quadrado do fator de potência.

$$S^2 = P^2 + Q^2 \quad 4.1$$

$$\cos \varphi = \frac{P}{Q} \quad 4.2$$

$$I = \frac{P_{aparente}}{\sqrt{3} \cdot V} = \frac{P_{ativa}}{\sqrt{3} \cdot V \cdot FP} \quad 4.3$$

A redução da potência aparente devido à redução da potência reativa pode ser observada na Figura 4.9, pois a potência reativa gerada nos barramentos 1, 2 e 3 diminui consideravelmente. Esta diminuição do fluxo de potência reativa esclarece a redução da percentagem de ocupação das linhas, visto que esta taxa diz respeito à potência nominal da linha. Em relação ao fluxo da potência ativa, sendo que houve uma redução de 3.722 MW nas perdas elétricas, essa redução é confirmada pela potência ativa gerada no barramento de referência, barramento 1. Na rede original, a potência gerada no barramento 1 era 108 MW, na rede já com a implementação dos bancos de condensadores a potência gerada no barramento 1 é 104.3 MW. A diferença da potência gerada no barramento 1 na rede original face à rede com implementação de bancos de condensadores corresponde à redução das perdas como esperado.

O perfil de tensões da rede original e da rede com a implementação dos bancos de condensadores correspondentes à melhor solução encontrada pelo algoritmo *PSO* e *EPSO* pode ser visto na Figura 4.10. Como se pode observar as tensões estão entre os limites toleráveis (0.95 pu a 1.05 pu). As tensões nos barramentos de geração (1, 2 e 3) mantiveram-se e nos barramentos de carga a tensão subiu. A implementação do banco de condensadores num barramento diminui a corrente e aumenta a tensão no barramento.

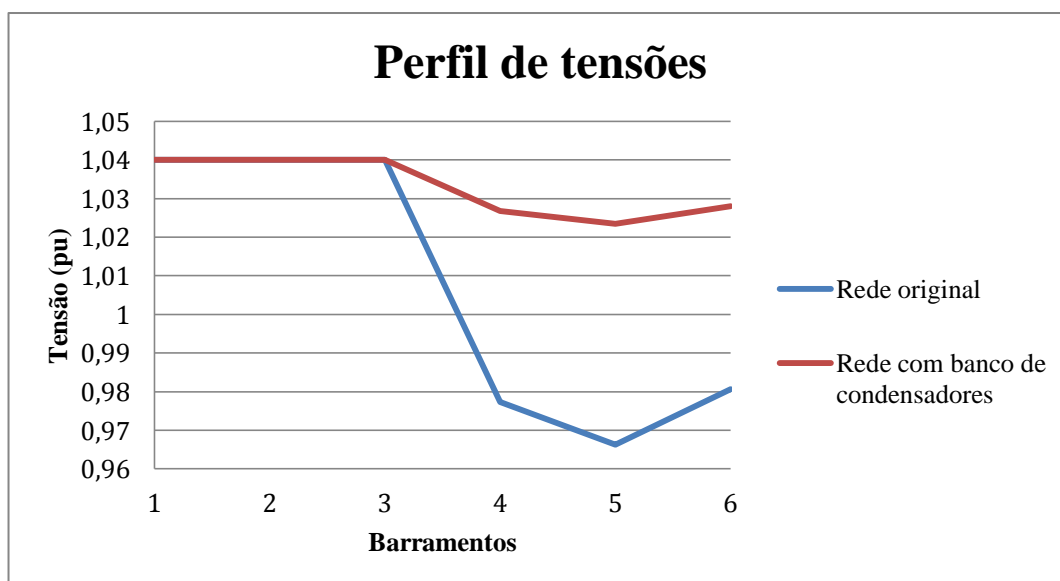


Figura 4.10: Perfil de tensões da rede original de seis barramentos e da mesma rede com implementação de condensadores.

4.4.2 Rede “IEEE14”

A rede que servirá como caso de estudo, baseada na rede de catorze barramentos [47], “IEEE14”, é apresentada na Figura 4.11. Foram feitos três testes para cada algoritmo. A configuração correspondente à melhor solução encontrada, que apresentou menores perdas para a rede elétrica de

catorze barramentos, foi a considerada para análise. Os dados da rede “IEEE14” estão apresentados no anexo 3, onde se constata que as perdas totais da rede original são 11.847 MW.

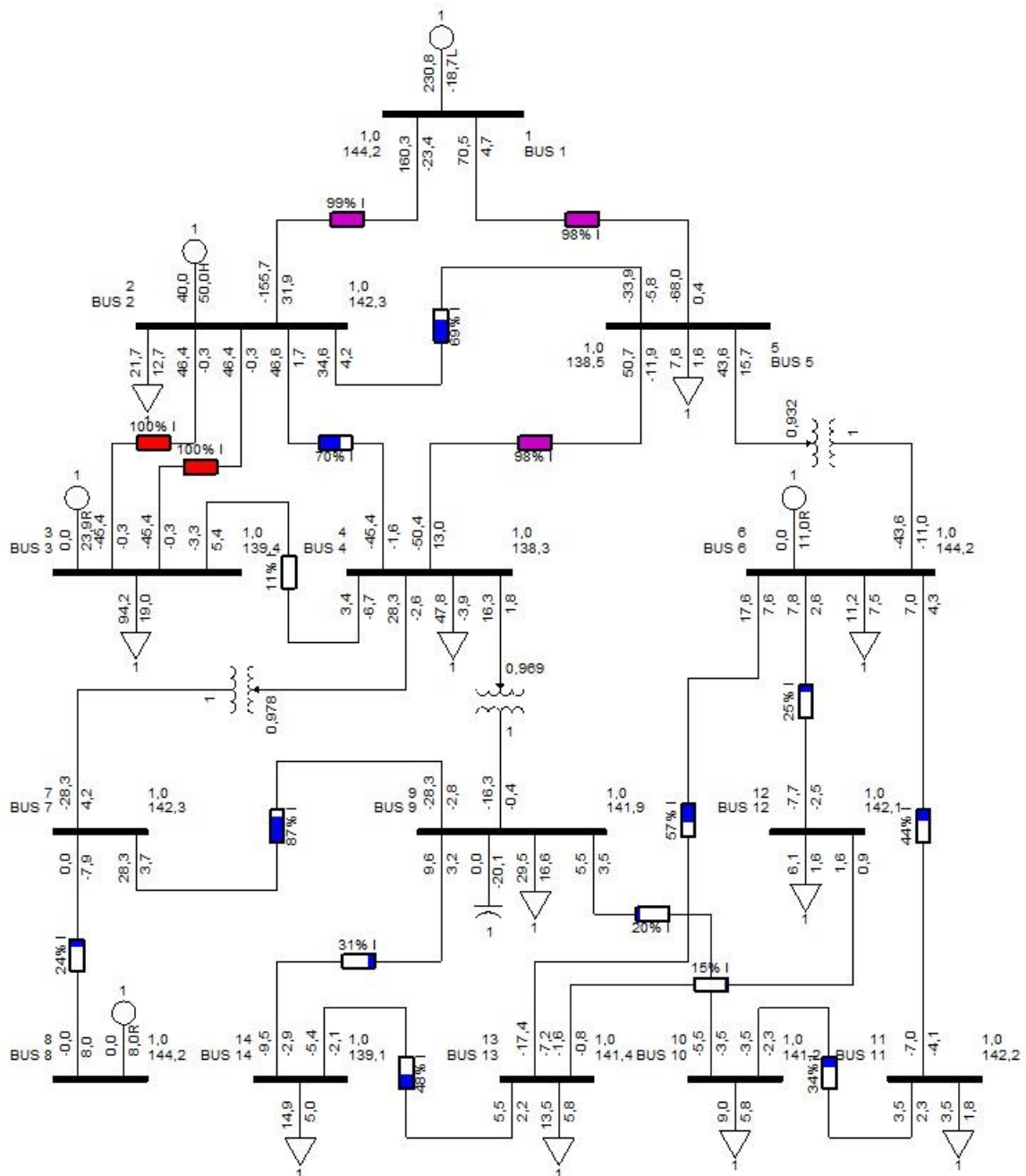


Figura 4.11: Esquema da rede original de catorze.

Aplicação do algoritmo *PSO*

Os parâmetros dimensionados para a rede de catorze barramentos referente ao algoritmo *PSO* estão apresentados na Tabela 13. Como foi referido anteriormente, foi aplicado três vezes o algoritmo *PSO* à rede de catorze barramentos. Conforme se pode verificar na Tabela 14 o terceiro teste foi o que apresentou as menores perdas elétricas totais (P_t).

Tabela 13: Parâmetros escolhidos para o algoritmo *PSO* quando aplicado à rede de catorze barramentos.

Partículas	Gerações	Capacidade Máxima	Capacidade Mínima	Velocidade Máxima	Velocidade Mínima
50	100	50	-50	10	-10

Tabela 14: Resultados obtidos das aplicações do algoritmo *PSO* à rede de catorze barramentos.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	11.694	-4	-11	-18	3	1	0	21	-15	10	5	1	1	5	5
2	11.688	-23	-12	-19	-3	1	7	13	0	16	5	-1	1	5	5
3	11.681	-8	-12	-17	-2	1	4	16	-19	16	5	1	1	5	5

Na Figura 4.12 é ilustrada a rede de catorze barramentos com a implementação dos bancos de condensadores correspondentes à configuração da melhor solução encontrada pelo algoritmo *PSO*.

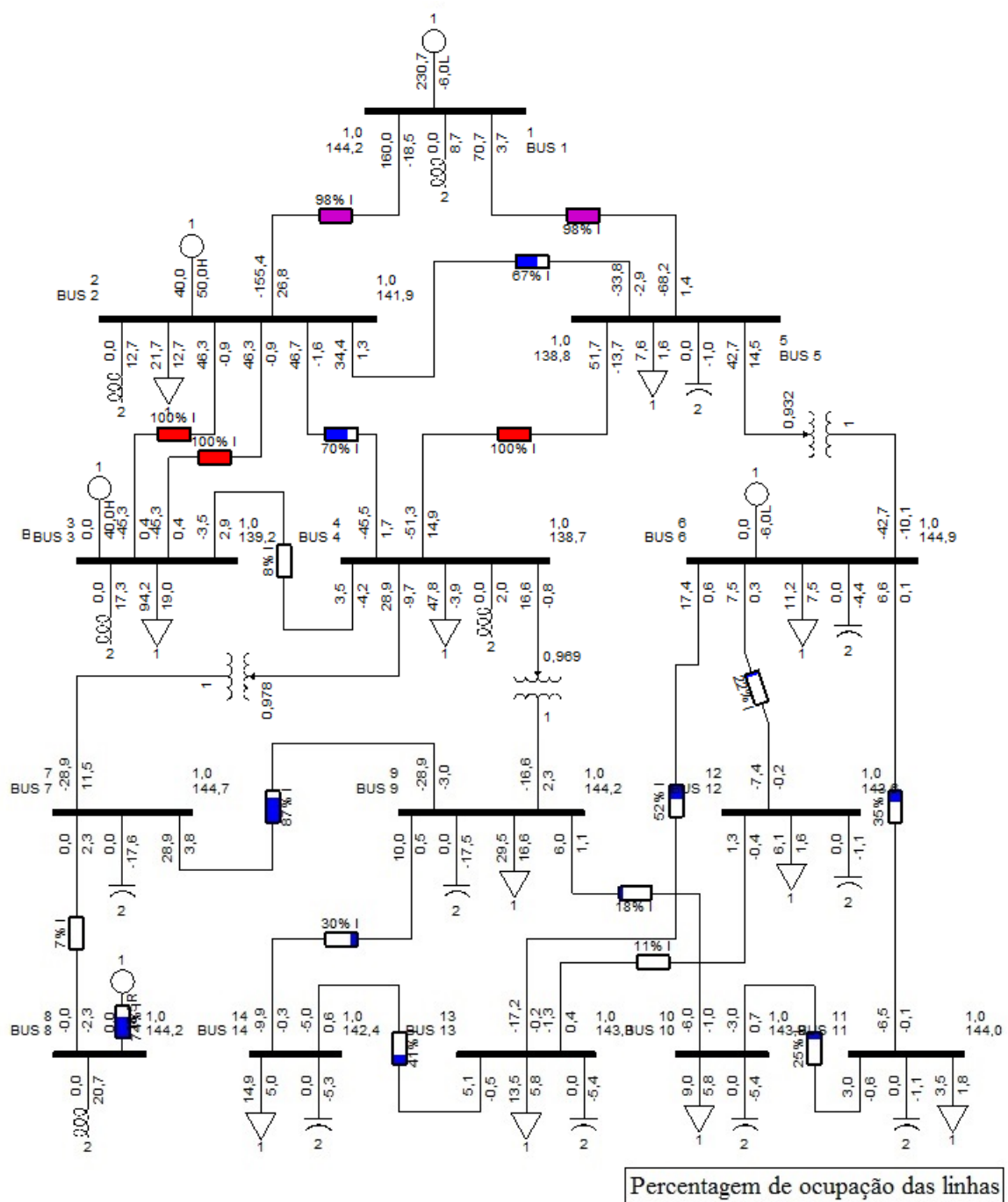


Figura 4.12: Esquema da rede de catorze barramentos depois de aplicado o algoritmo *PSO*.

Aplicação do algoritmo *EPSO*

Foram utilizados os mesmos parâmetros do algoritmo *PSO*, Tabela 13, na aplicação do algoritmo *EPSO*. Os resultados obtidos dos três testes feitos com o algoritmo *EPSO* à rede de catorze barramentos estão expostos na Tabela 15. O teste que apresentou menores perdas elétricas totais foi o segundo.

Tabela 15: Resultados obtidos das aplicações do algoritmo *EPSO* à rede de catorze barramentos.

Testes	Perdas totais [MW]	Capacidade [MVar] instalada por barramento													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	11,691	-30	-12	4	0	1	-1	23	-9	9	4	0	1	3	5
2	11,686	2	-12	4	2	1	-1	11	0	15	5	1	1	5	5
3	11,690	3	-12	3	2	1	3	16	-10	10	5	1	0	5	5

Na Figura 4.13 é ilustrada a rede de catorze barramentos com a implementação correspondente à configuração da melhor solução encontrada pelo algoritmo *EPSO*.

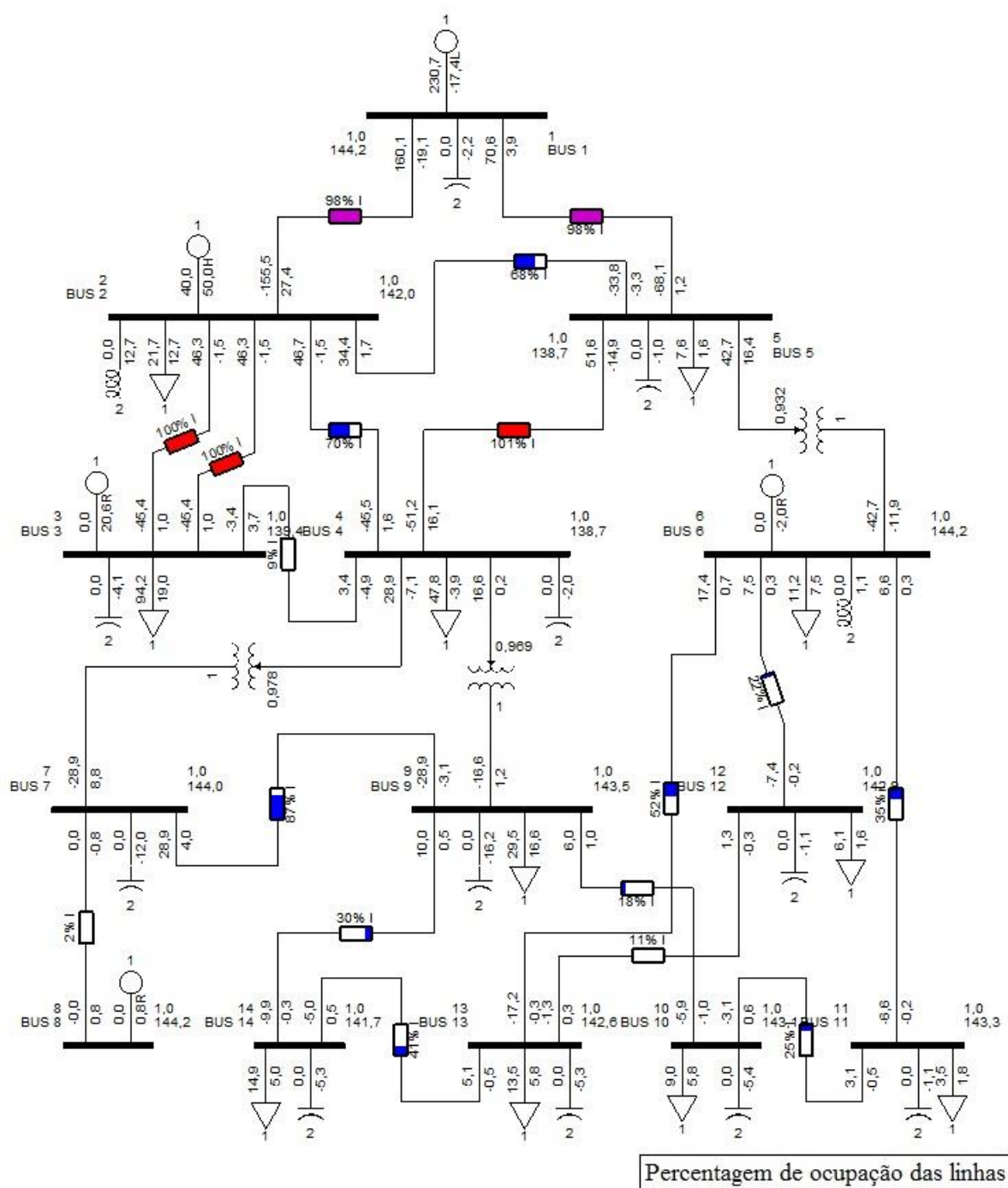


Figura 4.13: Esquema da rede de catorze barramentos depois de aplicado o algoritmo *EPSO*.

As perdas elétricas de todas as linhas e as perdas totais para o caso da rede original de catorze barramentos e para os casos da rede com implementação de bancos de condensadores estão apresentadas na tabela 2 do Anexo V.

Análise dos resultados

A rede de catorze barramentos original apresentava perdas elétricas totais iguais a 11.847 MW. A melhor solução encontrada pelo algoritmo *PSO* apresentou menores perdas totais relativamente à do algoritmo *EPSO*, uma diferença de 6 kW, como se pode constatar pela tabela 2 do Anexo IV. Em relação ao algoritmo *PSO*, os resultados obtidos mostram que as perdas elétricas totais da rede, depois de implementados os bancos de condensadores correspondentes à melhor solução encontrada, reduziram para 11.681 MW, o que significa uma redução de 1.4 %. Os resultados obtidos, quando aplicado o algoritmo *EPSO*, apontam para uma redução de 1.35 % das perdas elétricas totais da rede. É evidente que a redução das perdas elétricas totais da rede de catorze barramentos é muito pouco significativa.

Sendo esta uma rede que representa uma porção do sistema elétrico da América do norte (Centro Oeste) no ano 1962, o perfil de carga, como se podem notar no anexo 3, é baixo, o que limita a capacidade máxima dos bancos de condensadores, pois uma das restrições da rede é não haver barramentos sobrecompensados. Na melhor solução encontrada tanto pelo algoritmo *PSO* como pelo *EPSO* é possível constatar que nos últimos barramentos da rede elétrica (barramentos de carga) a capacidade dos bancos de condensadores a implementar é a máxima possível. Nos barramentos iniciais, os barramentos de geração (1, 2 e 3), pode observar-se, pela melhor solução encontrada por o algoritmo *PSO*, que a capacidade dos bancos de condensadores a implementar é negativa ou seja, em vez de serem condensadores são bobinas. Este facto deve-se ao excesso de potência reativa, de carácter capacitivo, que os geradores nos barramentos 2 e 3 estão a enviar para a rede. Na melhor solução encontrada pelo algoritmo *EPSO*, já não acontece o mesmo, isto deve-se à redução de potência reativa enviada para a rede do gerador no barramento 3 face à solução do algoritmo *PSO*. Desta forma, no barramento 3, a solução sugere a implementação de um banco de condensadores que irá suprimir a potência reativa de carácter indutivo no barramento.

Analisando, então, os barramentos importantes para a redução de perdas da rede, pode afirmar-se que os barramentos 7 e 9 são os barramentos onde foi implementado os bancos de condensadores com maior capacidade. Como se percebe pela tabela 2 do Anexo IV, não houve nenhuma linha que visse as suas perdas elétricas serem reduzidas significativamente. Isto pode ser explicado, pois a injeção de potência reativa na rede continuou igual, tanto para a nova configuração da rede correspondente à melhor solução encontrada pelo algoritmo *PSO*, Figura 4.12, como para o algoritmo *EPSO*, Figura 4.13. Desta forma, não houve uma grande melhoria do fator de potência, o que se traduz numa fraca redução das perdas elétricas totais.

O perfil de tensões da rede elétrica original de catorze barramentos face à rede elétrica de catorze barramentos com implementação de bancos de condensadores correspondentes à melhor solução encontrada pelo algoritmo *PSO* e pelo algoritmo *EPSO* podem ser observados na Figura 4.14 e Figura 4.15, respetivamente. A rede original não apresentava nenhum barramento com tensão abaixo da tensão nominal, o que é importante para o bom funcionamento da rede. Nos barramentos onde foram implementados bancos de condensadores a tensão aumentou, nos barramentos onde foram implementadas bobinas a tensão desceu.

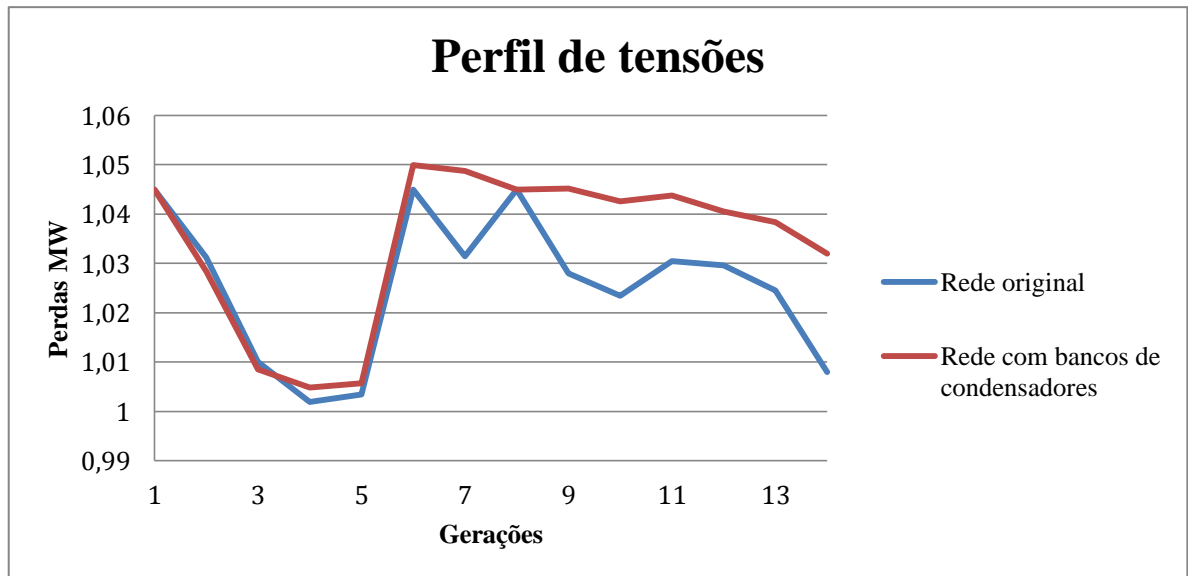


Figura 4.14: Perfil de tensões da rede original de catorze barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo *PSO*.

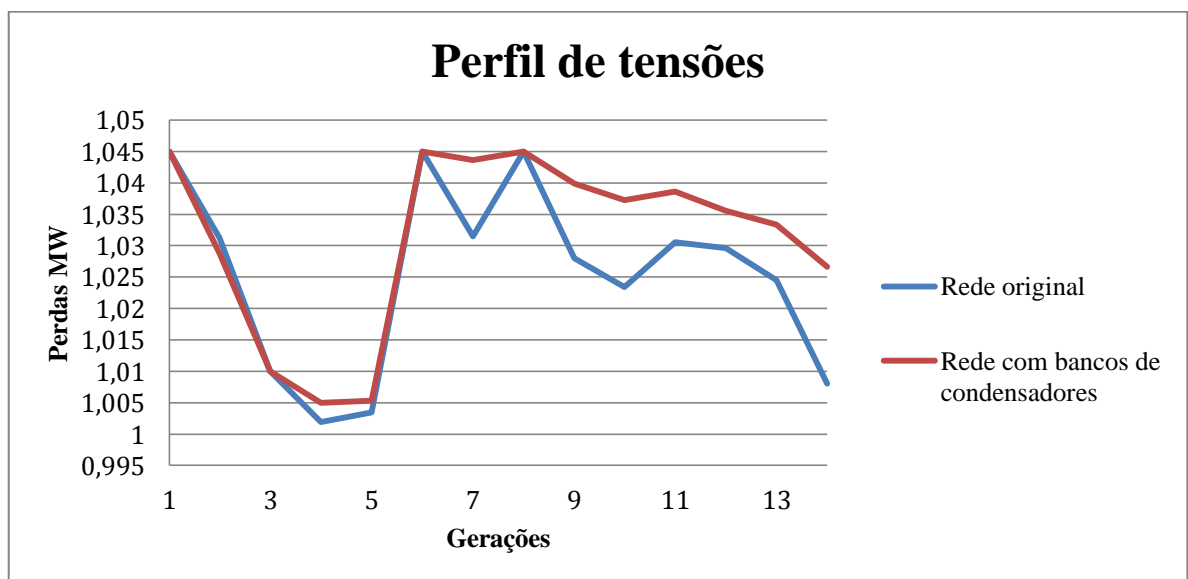


Figura 4.15: Perfil de tensões da rede original de catorze barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo *EPSO*.

4.4.3 Rede de vinte e um barramentos

A rede de vinte e um barramentos, “Rede_MAT_AT”, é apresentada na Figura 4.16. Os dados da rede podem ser vistos no anexo 4, onde se observa que as perdas totais da rede original são 10.154 MW. A rede representa uma topologia típica de produção, transporte e distribuição de energia elétrica. Esta rede é constituída por três níveis de tensão:

- Nos primeiros três barramentos estão localizadas as centrais produtoras de energia elétrica;
- A tensão elétrica é elevada nas subestações e transportada nos seguintes quatro barramentos até as áreas de consumo;
- Perto das áreas de consumo, nas subestações é reduzido o nível da tensão para a eletricidade poder ser distribuída, em média tensão, nos últimos catorze barramentos.

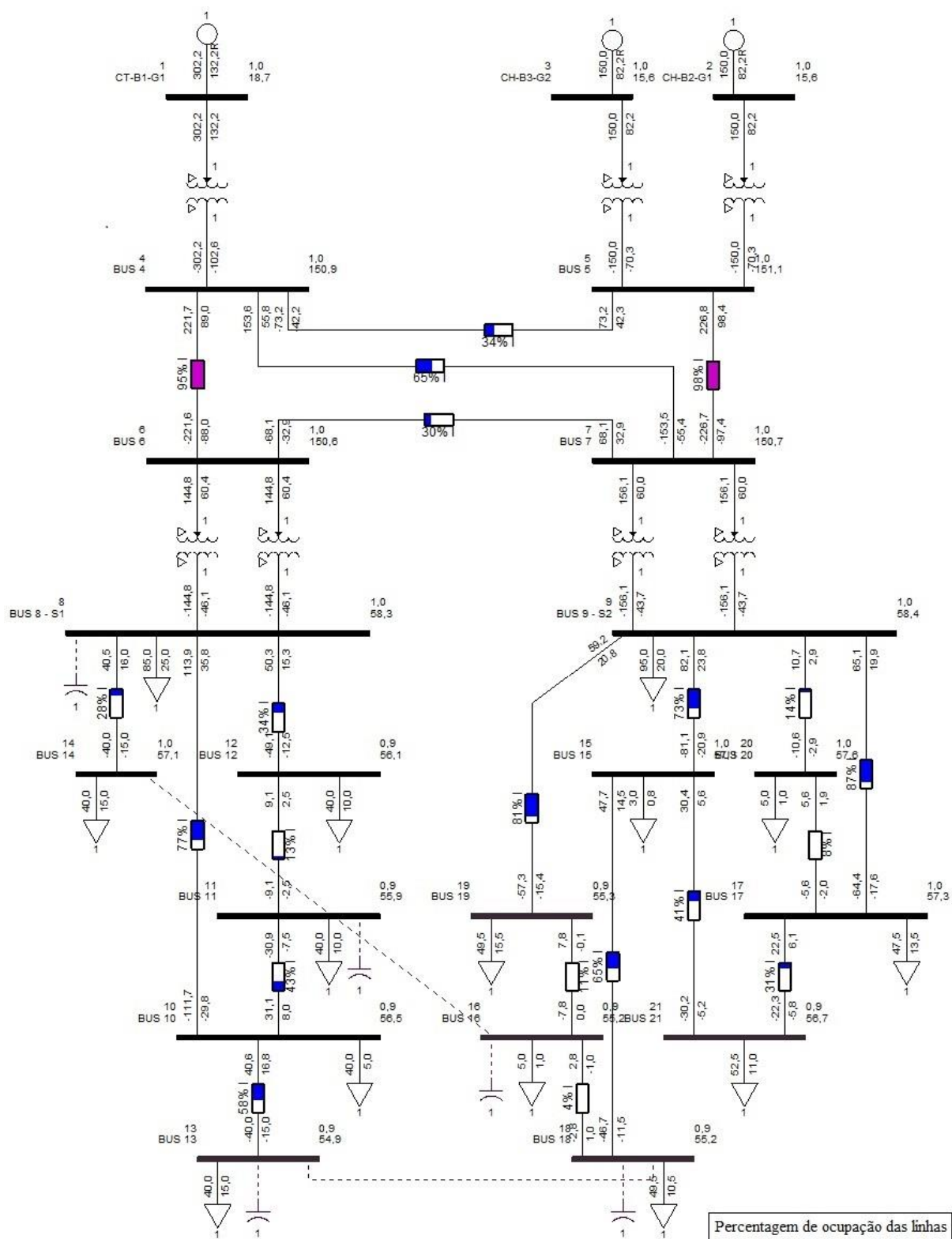


Figura 4.16: Esquema da rede original de vinte e um barramentos.

Aplicação do algoritmo *PSO*

Os parâmetros dimensionados para a rede de vinte e um barramentos referente ao algoritmo *PSO* são apresentados na Tabela 16. Devido às múltiplas combinações possíveis de soluções, o número de gerações foi aumentado em relação aos casos de estudo anteriormente analisados, assim a convergência para a solução ideal é garantida.

Tabela 16: Parâmetros escolhidos para o algoritmo *PSO* quando aplicado à rede de vinte e um barramentos.

Partículas	Gerações	Capacidade	Capacidade	Velocidade	Velocidade
		Máxima	Mínima	Máxima	Mínima
50	120	150	-50	10	-10

O segundo teste revelou as menores perdas totais, como se pode verificar na Tabela 17.

Tabela 17: Resultados obtidos das aplicações do algoritmo *PSO* à rede de vinte e um barramentos.

Teste 1	Bus	1	2	3	4	5	6	7	8	9	10	11
	Cap. [MVar]	15	18	-23	53	54	21	51	11	14	2	10
	Bus	12	13	14	15	16	17	18	19	20	21	P _t
Teste 2	Cap. [MVar]	6	14	15	0	1	13	10	15	1	11	7.977
	Bus	1	2	3	4	5	6	7	8	9	10	11
	Cap. [MVar]	30	50	0	59	8	35	53	25	20	5	10
Teste 3	Bus	12	13	14	15	16	17	18	19	20	21	P _t
	Cap. [MVar]	10	15	15	0	1	7	10	15	-1	11	7.920
	Bus	1	2	3	4	5	6	7	8	9	10	11
Teste 4	Cap. [MVar]	0	31	36	47	20	40	57	12	20	5	10
	Bus	12	13	14	15	16	17	18	19	20	21	P _t
	Cap. [MVar]	10	15	15	0	1	13	10	14	1	9	7.930

Na Figura 4.17 é apresentado o esquema da rede de vinte e um barramentos com a implementação dos bancos de condensadores correspondentes à configuração da melhor solução encontrada pelo algoritmo *PSO*.

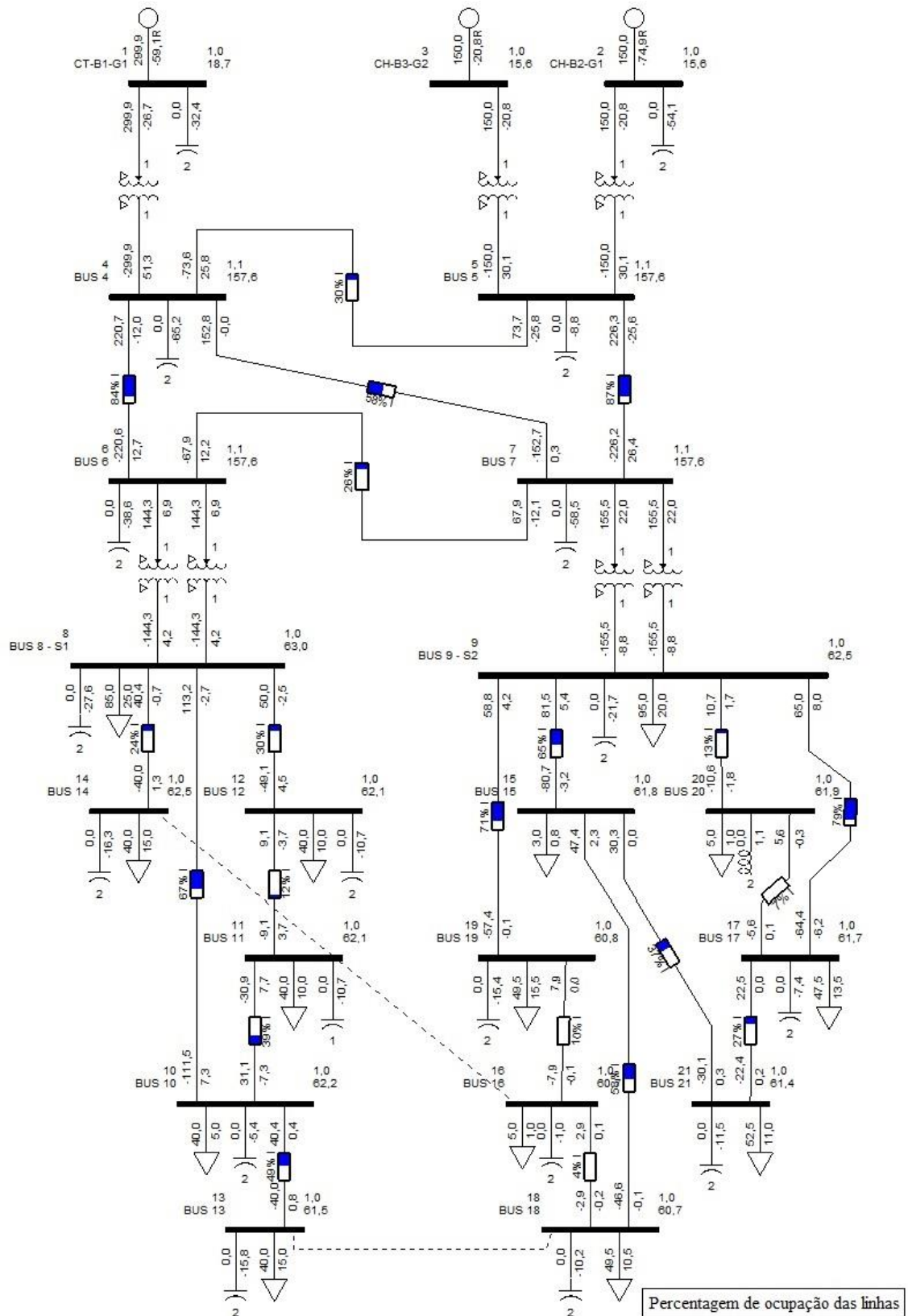


Figura 4.17: Esquema da rede de vinte e um barramentos depois de aplicado o algoritmo *PSO*.

Aplicação do algoritmo *EPSO*

Os parâmetros que serviram para o algoritmo *PSO* também se adequam ao algoritmo *EPSO*, por conseguinte, os parâmetros usados na aplicação do algoritmo *EPSO* estão apresentados na Tabela 16. Os resultados obtidos dos três testes feitos, aplicando o algoritmo *EPSO*, estão expostos na Tabela 18. O teste que apresentou menores perdas elétricas totais foi o primeiro.

Tabela 18: Resultados obtidos das aplicações do algoritmo *EPSO* à rede de vinte e um barramentos.

Teste 1	Bus	1	2	3	4	5	6	7	8	9	10	11
	Cap.	32	-1	28	22	42	62	32	25	16	5	10
	[MVar]											
	Bus	12	13	14	15	16	17	18	19	20	21	P_t
Teste 2	Cap.	9	15	14	0	1	12	9	15	0	10	7.919
	[MVar]											
	Bus	1	2	3	4	5	6	7	8	9	10	11
	Cap.	41	61	-8	5	2	48	91	25	19	3	8
Teste 3	[MVar]											
	Bus	12	13	14	15	16	17	18	19	20	21	P_t
	Cap.	10	15	15	0	1	9	10	14	1	11	7.943
	[MVar]											
Teste 3	Bus	1	2	3	4	5	6	7	8	9	10	11
	Cap.	17	26	6	7	33	66	55	24	20	5	10
	[MVar]											
	Bus	12	13	14	15	16	17	18	19	20	21	P_t
Teste 3	Cap.	8	15	13	0	0	13	8	15	-1	8	7.936
	[MVar]											

Na Figura 4.18 é apresentada a rede de vinte e um barramentos com a implementação correspondente à configuração da melhor solução encontrada pelo algoritmo *EPSO*.

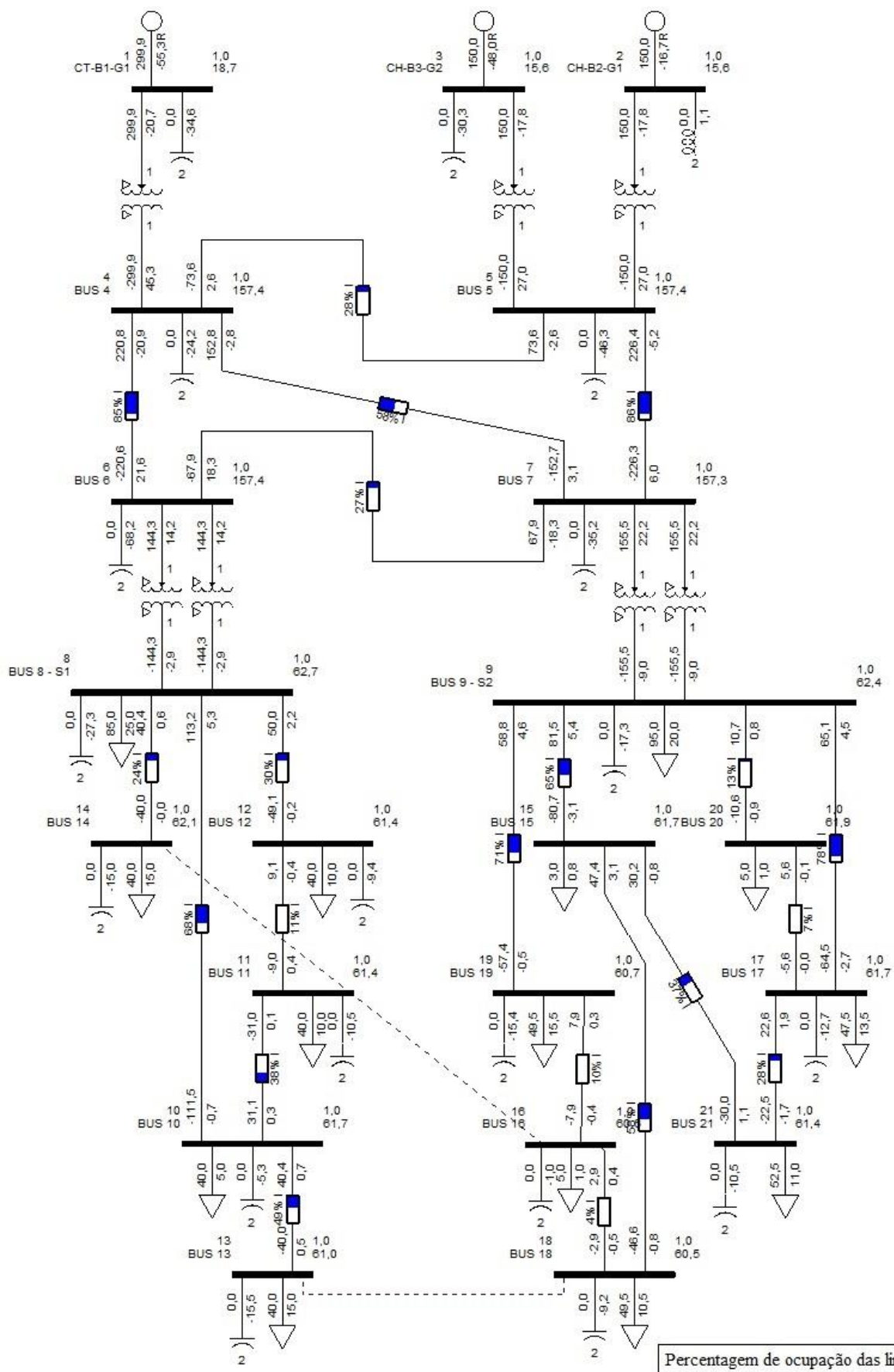


Figura 4.18: Esquema da rede de vinte e um barramentos depois de aplicado o algoritmo *EPSO*.

As perdas elétricas de todas as linhas e as perdas totais para o caso da rede original de vinte e um barramentos e para o caso da mesma rede com implementação de bancos de condensadores estão expostas na tabela 3 do anexo IV.

Análise dos resultados

A rede de vinte e um barramentos original exibiu um total de 10.154 MW de perdas elétricas. Ambas as melhores soluções encontradas pelos algoritmos *PSO* e *EPSO* apresentaram uma redução de cerca de 22%. As perdas elétricas totais da rede com implementação dos bancos de condensadores, correspondentes à melhor solução encontrada pelo algoritmo *PSO*, foram 7.920 MW e pelo algoritmo *EPSO*, foram 7.919.

Em relação à localização e capacidade dos bancos de condensadores na rede de vinte e um barramentos, é notório que, nos barramentos onde não existe cargas (1 a 7), a capacidade dos bancos de condensadores instalados é grande. Como se pode observar na Figura 4.16, nas várias linhas que ligam os barramentos 1 a 9 há uma grande circulação de potência reativa. Com a implementação de bancos de condensadores, nestes barramentos ocorre uma grande redução da potência reativa, diminuindo assim a corrente que circula nessas linhas, o que significa uma redução da potência aparente e melhoria do fator de potência. Os geradores que na rede original eram responsáveis por tentarem compensar o fator de potência produzindo potência reativa, depois de implementados os bancos de condensadores na rede elétrica, os geradores passaram a absorver potência reativa, como se pode constatar na diferença entre Figura 4.16 face às Figura 4.17 e Figura 4.18. Nos barramentos de carga, a capacidade dos bancos de condensadores implementados é a suficiente para suprimir a potência reativa libertada pelas cargas nos barramentos.

A diferença das melhores soluções encontradas pelos dois algoritmos passa pela distribuição da capacidade dos bancos de condensadores em determinados barramentos. Em relação ao primeiro nível de tensão (barramentos 1, 2 e 3), a solução devolvida pelo algoritmo *PSO* mostra que a implementação dos bancos de condensadores deve ser feita nos barramentos 1 e 2, ao passo que a solução devolvida pelo algoritmo *EPSO* mostra que a implementação dos bancos de condensadores deve ser feita nos barramentos 1 e 3. No que diz respeito ao segundo nível de tensão (barramentos 4, 5, 6 e 7), o algoritmo *PSO* mostra que nos barramentos 4 e 7 se deve concentrar a maior capacidade dos bancos de condensadores, ao invés do algoritmo *EPSO* que mostra uma distribuição mais uniforme. Contudo, se somarmos as capacidades dos bancos de condensadores deste nível, a soma é idêntica para as duas soluções. Na solução do algoritmo *PSO* o total é 155 MVar, e na solução do algoritmo *EPSO* o total é 158. Tanto a solução do algoritmo *PSO* como a do algoritmo *EPSO* mostram os mesmos valores para os bancos de condensadores a implementar nos barramentos do terceiro nível de tensão.

Apesar do objetivo do problema *OCP* não ser a redução de linhas em sobrecarga da rede elétrica, esse facto acontece por via da otimização da rede. Como se pode constatar, embora a rede original não apresente linhas em sobrecarga, apenas duas linhas estavam perto do limite máximo admissível, na rede com implementação de condensadores, a percentagem de ocupação das linhas diminui, sendo que as duas linhas que estavam perto do limite máximo admissível deixaram de o estar.

No que diz respeito às tensões nos barramentos, a melhoria do perfil de tensões é bastante significativa. Como se pode verificar pela Figura 4.19 e Figura 4.20, a qualidade do fornecimento de energia elétrica às cargas (consumidores) na rede original é fraca, dado que os barramentos de 8 a 21 apresentam tensões abaixo da nominal. Com a implementação dos bancos de condensadores, a tensão desses barramentos subiu consideravelmente, permitindo assim que as cargas nestes barramentos funcionem sem falhas e que nestes barramentos seja possível o aumento da carga.

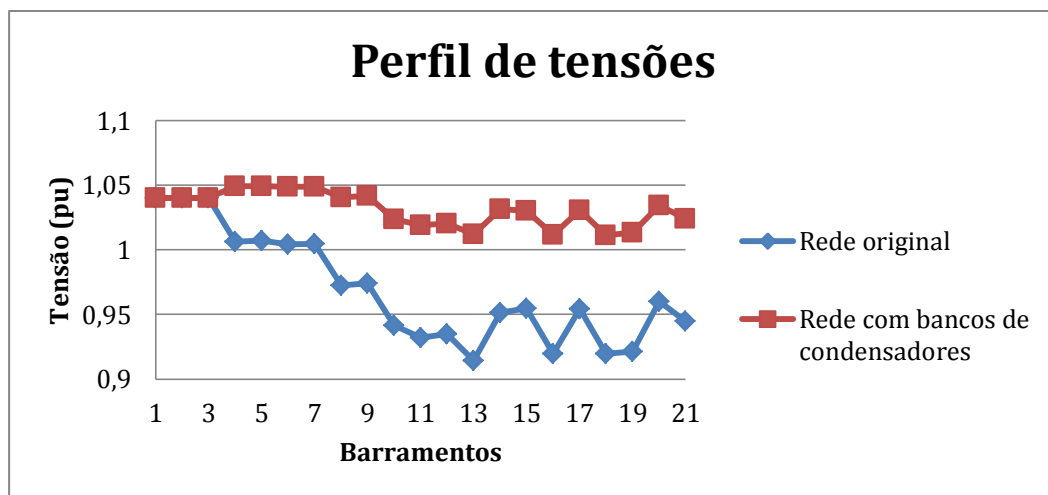


Figura 4.19: Perfil de tensões da rede original de vinte e um barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo *PSO*.

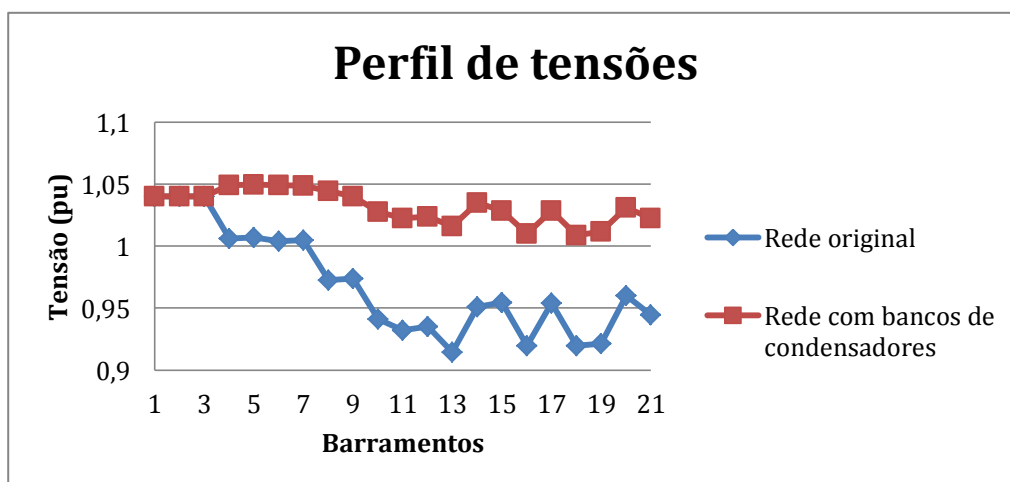


Figura 4.20: Perfil de tensões da rede original de vinte e um barramentos e da mesma rede com implementação de condensadores correspondente à melhor solução encontrada pelo algoritmo *EPSO*.

4.5 Eficiência e Robustez dos algoritmos

No decorrer do trabalho desenvolvido, até à última versão dos algoritmos *PSO* e *EPSO*, foram testadas diferentes abordagens para solucionar o problema *OCP*. A função que avaliava cada partícula foi sempre a que necessitou de mais análise, suscitando assim constantes alterações. Devido a esta ocorrência, outros pormenores do algoritmo foram postos em segundo plano.

Uma das características que se revelou muito significativa na qualidade dos resultados obtidos da aplicação de qualquer um dos algoritmos foi o tamanho do espaço de busca da partícula. Inicialmente, o tamanho (capacidade do banco de condensadores) do espaço de busca de qualquer dimensão (barramento) era igual para todas as dimensões, ou seja, o valor da capacidade do banco de condensadores estava compreendido entre as constantes, capacidade máxima e mínima, estipuladas pelo utilizador. Caso algum barramento estivesse sobrecompensado (a capacidade do banco de condensadores de um barramento é maior que a potência reativa da carga desse mesmo barramento) seriam aplicadas penalidades à avaliação da partícula. Estas penalidades representavam um acréscimo no valor da aptidão da partícula.

A razão que fundamentava este ponto de vista era uma das essências do algoritmo *PSO*, que é a partilha de experiências entre todas as partículas da população. Quando uma partícula fosse penalizada por sobrecompensação essa informação era partilhada. Esta ideia foi posta de parte e o espaço de busca limitado apenas às capacidades possíveis em determinados barramentos. Embora, através destes testes, ficasse provado que o algoritmo *EPSO* com o acrescentar de restrições conseguia resultados satisfatórios ao invés do algoritmo *PSO*.

4.5.1 Rede de catorze barramentos

Ambos os algoritmos, nas condições do parágrafo anterior, foram aplicados cinquenta vezes à rede de catorze barramentos e à rede de vinte e um barramentos. Na Figura 4.21 são apresentados os cinquenta testes referentes à aplicação do algoritmo *PSO* na rede de catorze barramentos. Dos cinquenta testes, apenas trinta apresentaram perdas totais inferiores às perdas totais da rede original. Nos restantes vinte testes, as melhores soluções encontradas pelo algoritmo *PSO* apresentavam barramentos em sobrecompensação revelando-se assim soluções inválidas.

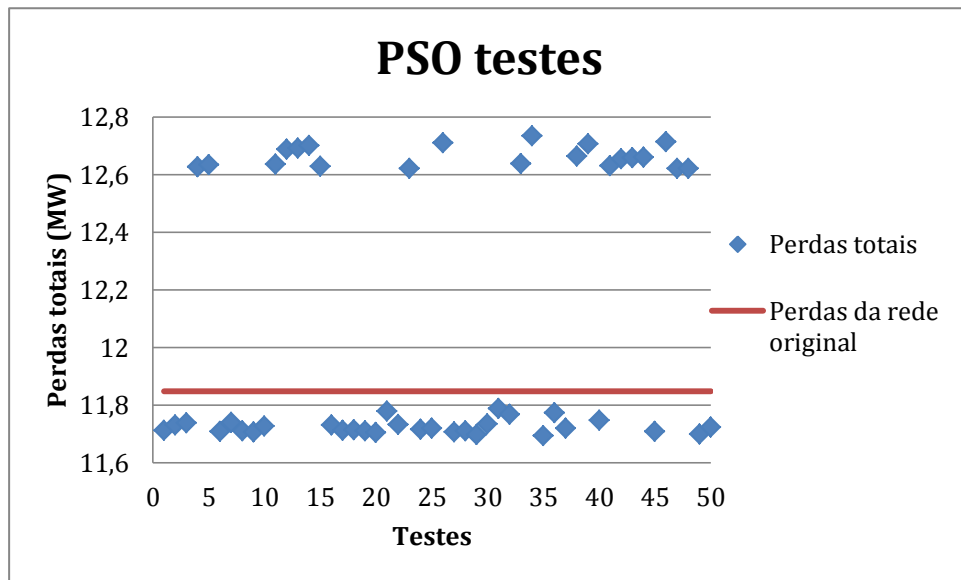


Figura 4.21: Eficiência e robustez do algoritmo *PSO* relativamente à rede de catorze barramentos.

Por sua vez, como se pode ver na Figura 4.22, em todos os cinquenta testes do algoritmo *EPSO*, a melhor solução encontrada apresentou sempre perdas totais menores que as perdas totais da rede original.

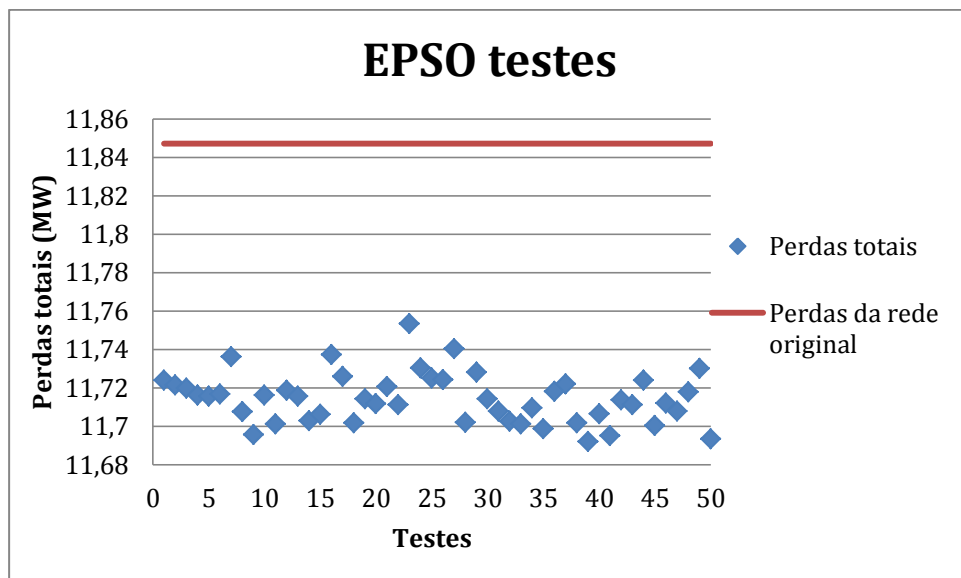


Figura 4.22: Eficiência e robustez do algoritmo *EPSO* relativamente à rede de catorze barramentos.

4.5.2 Rede de vinte e um barramentos

O algoritmo *PSO* revelou-se incapaz de resolver o problema da *OCP* para a rede de vinte e um barramentos. Como se pode observar na Figura 4.23, em todos os testes efetuados, a melhor solução encontrada pelo algoritmo *PSO* é uma solução inválida.

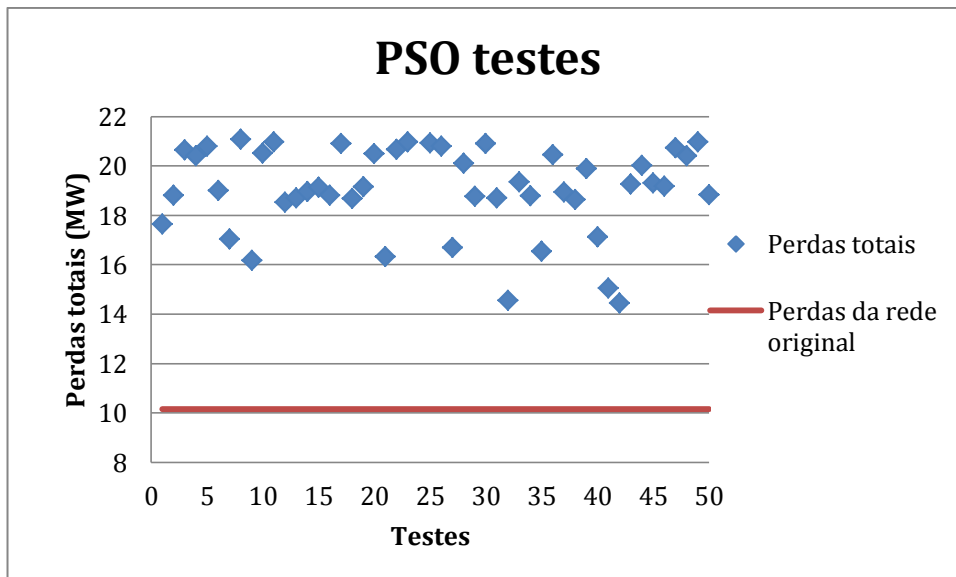


Figura 4.23: Eficiência e robustez do algoritmo *PSO* relativamente à rede de vinte e um barramentos.

A melhor solução encontrada pelo algoritmo *EPSO*, nos cinquenta testes efetuados, apresentou, quase sempre, perdas elétricas totais menores que as perdas elétricas totais da rede original. Apenas quatro testes apresentaram resultados negativos.

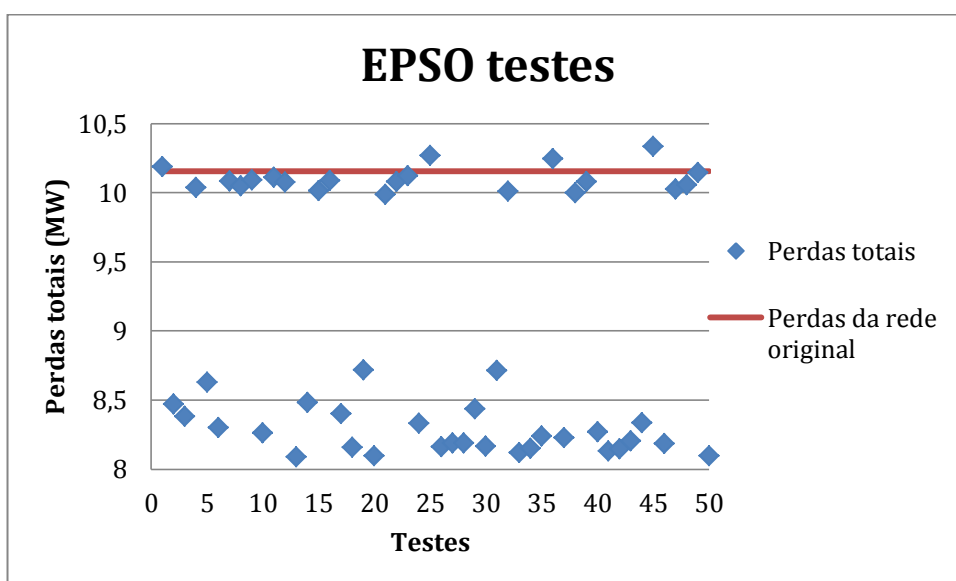


Figura 4.24: Eficiência e robustez do algoritmo *EPSO* relativamente à rede de vinte e um barramentos.

Através desta experiência, ficou provado que, em trabalhos futuros, com a inclusão de novos módulos no programa, o algoritmo *PSO* pode revelar-se inapto a resolver o problema *OCP*. O algoritmo *EPSO*, por sua vez, devido aos fatores de mutação serem adaptativos, consegue auto ajustar-se e as partículas da população convergirem para a solução ótima.

Foi analisada a alteração dos pesos dos parâmetros estratégicos de ambos algoritmos para um teste referente à rede de vinte e um barramentos. Na Figura 4.25, pode observar-se a diferença entre o parâmetro estratégico inércia adotado para o algoritmo *PSO* face ao algoritmo *EPSO*. Como se pode constatar para o algoritmo *PSO* o parâmetro é constante. Para o algoritmo *EPSO*, o parâmetro, que é o parâmetro inércia da melhor partícula encontrado em cada geração, tende a diminuir. Esta ocorrência deve-se ao facto da partícula já estar num local sub ótimo do espaço de busca e a velocidade da partícula precisar de ser cada vez menor para que a partícula consiga achar a solução ideal.

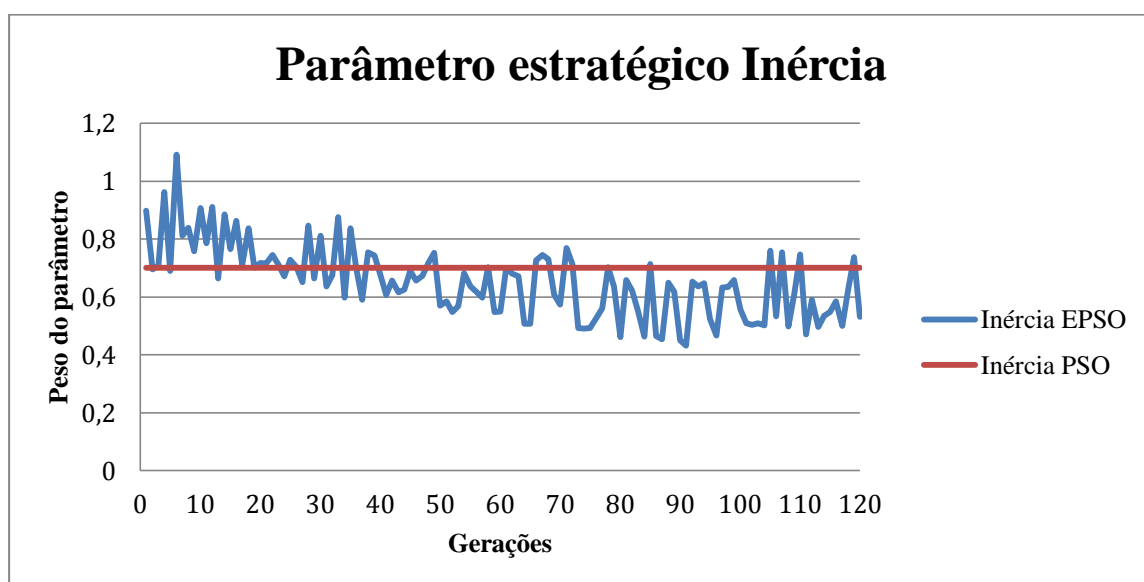


Figura 4.25: Diferença entre o peso do parâmetro estratégico inércia do algoritmo *PSO* e *EPSO* com o avançar das gerações.

Na Figura 4.26 é ilustrado a alteração dos pesos dos parâmetros estratégicos memória e cooperação do algoritmo *EPSO*. Nesta figura não são expostos os pesos dos parâmetros memória e cooperação do algoritmo *PSO*, visto que o peso para qualquer parâmetro é sempre um número aleatório entre dois e zero. Mais uma vez é elucidado que o peso dos parâmetros tende a diminuir, devido ao facto explicado no parágrafo anterior.

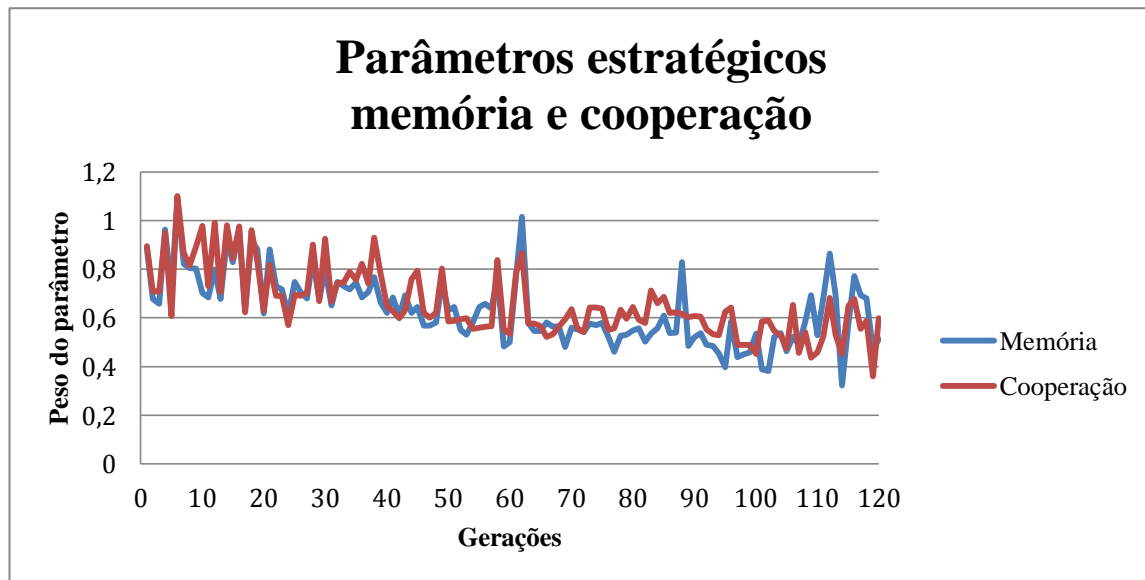


Figura 4.26: O peso dos parâmetros estratégicos memória e cooperação do algoritmo *EPSO* com o avançar das gerações.

5. Conclusão

5.1 Observações finais

A escolha do tema redução de perdas elétricas numa rede elétrica recaiu sobre o facto de este tema ser a base para o aumento da eficiência energética num sistema de energia elétrica. A solução para o problema de entre muitas possíveis consistia na compensação de energia reativa através de bancos de condensadores. O problema que se pretendeu resolver, no decorrer deste trabalho, foi encontrar os locais (barramentos) numa dada rede elétrica onde implementar bancos de condensadores, com a capacidade ideal, para minimizar as perdas elétricas totais dessa rede. O objetivo proposto no início da dissertação foi desenvolver uma metodologia que fosse capaz de resolver esse problema, o problema *OCP*, para qualquer rede elétrica.

A metodologia adotada desenvolvida no capítulo 3, apresenta o algoritmo *PSO* e o *EPSO*. Estes foram os algoritmos utilizados para resolver o problema, sendo que o algoritmo *EPSO* é uma evolução do algoritmo *PSO*. Foram escolhidos estes dois algoritmos com a finalidade de analisar se o algoritmo *EPSO* apresentava melhores resultados que o algoritmo *PSO*.

No capítulo 4 são apresentados os resultados referentes à aplicação de ambos os algoritmos para três diferentes redes. As três redes que serviram de modelos de teste foram redes de seis, catorze e vinte e um barramentos. Dada a natureza combinatória do problema, quanto mais barramentos a rede tem, maior é o número de combinações possíveis para solucionar o problema da *OCP* dessa rede.

Constata-se que ambos os algoritmos foram capazes de resolver o problema *OCP* para as redes que serviram de teste. Apesar das alterações (tamanho do espaço de busca) que foram feitas ao algoritmo *PSO* e *EPSO* para que conseguissem resolver o problema *OCP* e apresentarem resultados satisfatórios, para trabalhos futuros, o algoritmo *PSO* pode não ser uma boa opção. Como ficou provado, caso sejam adotadas mais restrições à rede ou outros critérios para a avaliação da partícula, aumentado assim a dificuldade em encontrar soluções (partículas) aptas, o algoritmo *PSO* vai apresentar resultados menos satisfatórios que o algoritmo *EPSO*.

Depois de analisados os resultados, conclui-se que tanto o algoritmo *PSO* como o algoritmo *EPSO*, para a conjuntura do problema em questão, revelaram-se boas soluções técnicas. Ficou claro que um utilizador, ao querer aplicar qualquer um dos algoritmos a uma rede, tem de aplicar o algoritmo algumas vezes à rede para uma correta parametrização do algoritmo. Só assim o utilizador conseguirá obter os melhores resultados possíveis, ou seja, as menores perdas elétricas totais para essa rede.

5.2 Perspetivas para trabalhos futuros

No decorrer do trabalho efetuado e como o tema foi estudado sem partir de trabalhos anteriores, foram surgindo várias ideias que podiam ser incluídas na estrutura do algoritmo. Algumas sugestões para o desenvolvimento deste trabalho são agora apresentadas:

- A sugestão que mais se destaca devido à sua importância neste género de estudo é a inclusão do fator económico. Visto que o principal objetivo é melhorar a eficiência energética da rede e apesar de ser ponto assente que é necessário reduzir a emissão de poluentes, o fator económico desempenha um papel importantíssimo no ato de passar das simulações para a realidade. Assim, torna-se necessária a inclusão de um módulo, no programa, que fizesse a gestão económica para uma determinada solução. Este módulo era responsável por fazer o *trade-off* entre o custo da implementação dos bancos de condensadores (custo do banco de condensadores, mão de obra, etc) e o retorno que essa implementação trazia (redução de produção de energia elétrica);
- Nesta dissertação, os algoritmos apenas foram aplicados a diferentes redes e para um cenário operacional. Em trabalhos futuros, para cada rede podem ser estudados diferentes cenários de carga/geração;
- Dado que não existe, no mercado atual, todas as capacidades para bancos de condensadores, o utilizador poderia fazer uma base de dados das capacidades dos bancos de condensadores que estivessem disponíveis para serem implementados. Seriam feitas modificações no algoritmo para que este apenas apresentasse soluções com essas capacidades de bancos de condensadores.

Referências bibliográficas

- [1] REN, “Perdas na rede de transporte.” [Online]. Available: <http://www.erse.pt/pt/electricidade/actividadesdosector/transporte/Paginas/RNT-Perdas.aspx>. [Accessed: 27-Nov-2015].
- [2] E. e E. Distribuição, “Perdas nas redes de distribuição.” [Online]. Available: <http://www.erse.pt/pt/electricidade/actividadesdosector/distribuicao/Paginas/RND-Perdas.aspx>. [Accessed: 27-Nov-2015].
- [3] Eurelectric, “Eurelectric: Power Statistics 2010,” *Full Rep.*, p. 16, 2010.
- [4] J. V. Schmill, “Optimum Size and Location of Shunt Capacitors on Distribution Feeders,” in *IEEE Trans. on Power Apparatus and Systems*, vol. 85, 1965, pp. 825–832.
- [5] N. M. Neagle and D. R. Samson, “Loss Reduction from Capacitors Installed on Primary Feeders,” *AIEE Transactions*, part III, vol. 75, 1956, pp. 950–959.
- [6] H. Duran, “Optimum number, location, and size of shunt capacitors in radial distribution feeders, A dynamic programming approach,” in *IEEE Trans. Power Apparatus and Systems*, vol. 87, no. 9, 1968, pp. 1769–1774.
- [7] T. S. A. Salam, A. Y. Chikhani, and R. Hackam, “A new technique for loss reduction using compensating capacitors applied to distribution systems with varying load condition,” in *IEEE Trans. Power Delivery*, vol. 9, 1994, pp. 819–827.
- [8] M. Chis, M. M. A. Salama, and S. Jayaram, “Capacitor placement in distribution systems using heuristic search strategies,” in *Proc. Inst. Elect. Eng. Gen. Transm. Dist.*, vol. 144, no. 3, 1997, pp. 225–230.
- [9] D. Das, “Reactive power compensation for radial distribution networks using genetic algorithm,” *Int. J. Electr. Power Energy Syst.*, vol. 24, no. 7, pp. 573–581, 2002.

- [10] B. Rodrigues, P. Junior, and A. M. Cossi, "A Multiobjective Model for Distribution System Planning based on Tabu Search," *PowerTech (POWERTECH), 2013 IEEE Grenoble*, vol. 1, no. 6, pp. 16–20, 2013.
- [11] S. P. Singh and a. R. Rao, "Optimal allocation of capacitors in distribution systems using particle swarm optimization," *Int. J. Electr. Power Energy Syst.*, vol. 43, no. 1, pp. 1267–1275, 2012.
- [12] K. Prakash and M. Sydulu, "Particle Swarm Optimization Based Capacitor Placement on Radial Distribution Systems," *2007 IEEE Power Eng. Soc. Gen. Meet.*, 2007.
- [13] D. Kaur and J. Sharma, "Multiperiod shunt capacitor allocation in radial distribution systems," *Int. J. Electr. Power Energy Syst.*, vol. 52, no. 1, pp. 247–253, 2013.
- [14] S. F. Mekhamer, H. M. Khattab, and a. M. a. Mahmoud, "Solution of the Capacitor allocation problem in distribution feeders considering load variation: A modified simulated annealing based approach," *2006 Elev. Int. Middle East Power Syst. Conf.*, vol. 1, pp. 302–309, 2006.
- [15] S. Farook and P. S. Raju, "Metaheuristic Algorithms for Capacitor Siting and Sizing to Improve Voltage Profile," vol. 5, no. 1, pp. 1208–1215, 2014.
- [16] H.-C. Chang and C.-C. Kuo, "Network reconfiguration in distribution systems using simulated annealing," in *Electric Power Systems Research*, vol. 29, no. 3, 1994, pp. 227–238.
- [17] J. P. Sucena Paiva, "Redes de Energia Eléctrica: Uma Análise Sistémica," IST Press, 2005, pp. 250–251.
- [18] C. Blum and A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.
- [19] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Nat. Comput.*, vol. 8, no. 2, pp. 239–287, 2009.
- [20] "Vladimiro Miranda." [Online]. Available: http://sigarra.up.pt/feup/pt/func_geral.formview?P_CODIGO=208389. [Accessed: 08-Sep-2015].
- [21] M. Melanie, "An introduction to genetic algorithms," *Computers & Mathematics with Applications*, vol. 32, no. 6, p. 133, 1996.

- [22] M. Dianati, I. Song, and M. Treiber, "An introduction to genetic algorithms and evolution strategies," *Univ. Waterloo, Canada*, 2002.
- [23] V. Miranda, "Computação Evolucionária: Uma introdução," pp. 0–73, 2005.
- [24] Z. Michalewicz, *genetic algorithms + data structures = evolution programs*. 1992.
- [25] H.-G. Beyer, H.-G. Beyer, H.-P. Schwefel, and H.-P. Schwefel, "Evolution strategies – A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3 – 52, 2002.
- [26] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. New York, 1966.
- [27] J. H. Holland, "Adaptation in Natural and Artificial Systems:," *Ann Arbor MI Univ. Michigan Press*, vol. Ann Arbor, 1975.
- [28] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, vol. 1, no. 1, pp. 31–39, 1995.
- [29] J. Kennedy and W. M. Spears, "Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," *1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell.* (, 1998.
- [30] J. Kennedy, "The particle swarm: social adaptation of knowledge," *Proc. 1997 IEEE Int. Conf. Evol. Comput. (ICEC '97)*, pp. 303–308, 1997.
- [31] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary Programming VII SE - 59*, San Diego, CA, 1998.
- [32] E. O. Wilson, "Man From Sociobiology to Sociology," *Sociobiology: The New Synthesis*. pp. 547–575, 1975.
- [33] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE Int. Conf.*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [34] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4. pp. 25–34, 1987.
- [35] F. H. Heppner and U. Grenande, "A stochastic nonlinear model for coordinated bird flocks," *S. Krasner, Ed., The Ubiquity of Chaos*. AAAS Publications, Washington, DC, 1990.

- [36] Y. Shi, "Particle Swarm Optimization," *IEEE Neural Networks Soc.*, no. February, pp. 8–13, 2004.
- [37] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell. (Cat. No.98TH8360)*, pp. 69–73, 1998.
- [38] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," *Evol. Comput. IEEE Trans.*, vol. 12, no. 2, pp. 171–195, 2008.
- [39] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.
- [40] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," *Proc. 2002 Congr. Evol. Comput. CEC 2002*, vol. 2, pp. 1671–1676, 2002.
- [41] V. Miranda and N. Fonseca, "EPSO - Best-Of-Two-Worlds Meta-Heuristic Applied To Power System Problems," *Evol. Comput. 2002. CEC '02. Proc. 2002 Congr.*, pp. 1080–1085, 2002.
- [42] V. Miranda and N. Fonseca, "New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control," *Proc. 14th Power Syst. Comput. Conf*, p. 6, 2002.
- [43] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," *1999 Evolutionary Computation*, Piscataway, NJ: IEEE Service Center, Washington, DC, pp. 1951–1957.
- [44] "Python a programming language." [Online]. Available: <https://www.python.org/>. [Accessed: 08-Sep-2015].
- [45] "Eclipse an integrated development environment." [Online]. Available: <https://eclipse.org/>. [Accessed: 08-Sep-2015].
- [46] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*, 2nd Editio. John Wiley and Sons Inc., New York, 1996.
- [47] R. Christie, "UW Power System Test Case Archive." [Online]. Available: <http://www.ee.washington.edu/research/pstca/>. [Accessed: 08-Sep-2015].

ANEXOS

Anexo I

Evolução da melhor partícula encontrada até determinada geração, aplicando o algoritmo PSO à rede de seis barramentos.

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
1	79	75	24	79	48	32	337
2	78	93	24	70	70	51	386
3	78	93	24	70	70	51	386
4	81	93	26	68	68	54	390
5	88	93	22	67	66	61	397
6	89	93	19	65	60	58	384
7	89	95	10	66	60	61	381
8	89	95	10	66	60	61	381
9	63	96	14	66	58	60	357
10	73	90	12	67	59	63	364
11	67	89	14	66	59	62	357
12	67	89	14	66	59	62	357
13	67	89	14	66	59	62	357
14	69	93	14	66	58	62	362
15	69	93	14	66	58	62	362
16	69	93	14	66	58	62	362
17	69	93	14	66	58	62	362
18	69	93	14	66	58	62	362
19	70	90	15	66	58	62	361
20	70	90	15	66	58	62	361
21	70	90	15	66	58	62	361
22	70	90	15	66	58	62	361
23	61	92	14	66	58	62	353
24	61	87	13	66	58	62	347

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
25	55	85	12	66	59	62	339
26	55	86	22	67	59	62	351
27	56	87	15	66	59	62	345
28	58	85	10	66	59	62	340
29	68	84	16	66	58	62	354
30	66	80	19	66	59	62	352
31	63	86	21	67	58	63	358
32	58	82	18	67	58	62	345
33	58	82	18	67	58	62	345
34	62	79	16	66	58	62	343
35	62	79	16	66	58	62	343
36	62	79	16	66	58	62	343
37	56	80	16	67	59	62	340
38	53	84	14	67	58	61	337
39	57	84	13	67	58	63	342
40	57	84	13	67	58	63	342
41	57	76	12	66	58	63	332
42	57	76	12	66	58	63	332
43	57	68	11	66	58	62	322
44	55	78	17	66	58	62	336
45	59	85	14	67	58	62	345
46	59	85	14	67	58	62	345
47	54	75	10	67	58	62	326
48	54	75	10	67	58	62	326
49	51	70	1	67	58	62	309
50	48	63	-3	67	58	62	301
51	46	58	-5	67	58	62	296
52	44	54	-7	67	58	62	292
53	39	56	0	67	58	62	282
54	37	55	0	67	58	62	279
55	37	55	0	67	58	62	279
56	39	50	-2	67	58	62	278

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
57	39	50	-2	67	58	62	278
58	33	49	-1	67	58	62	270
59	26	48	-2	67	58	62	263
60	21	47	-3	67	58	62	258
61	21	37	1	67	58	62	246
62	14	32	2	67	58	62	235
63	9	28	3	67	58	62	227
64	5	25	3	67	58	62	220
65	0	19	3	67	58	62	209
66	-5	15	1	67	58	62	208
67	-3	16	1	67	58	62	207
68	0	10	1	67	58	62	198
69	0	4	0	67	58	62	191
70	0	0	-1	67	58	62	188
71	0	0	-1	67	58	62	188
72	0	0	-1	67	58	62	188
73	0	0	-1	67	58	62	188
74	0	0	-1	67	58	62	188
75	0	0	-1	67	58	62	188
76	0	0	-1	67	58	62	188
77	0	0	-1	67	58	62	188
78	0	0	0	67	58	62	187
79	0	0	0	67	58	62	187
80	0	0	0	67	58	62	187

Evolução da melhor partícula encontrada até determinada geração, aplicando o algoritmo EPSO à rede de seis barramentos.

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
1	86	76	12	56	47	65	342
2	83	69	11	56	57	68	344
3	9	27	74	64	59	70	303
4	0	17	73	70	57	60	277
5	0	17	73	70	57	60	277
6	0	17	73	70	57	60	277
7	9	12	72	67	56	64	280
8	9	12	72	67	56	64	280
9	9	12	72	67	56	64	280
10	20	-7	65	68	59	62	281
11	20	-7	65	68	59	62	281
12	20	-7	65	68	59	62	281
13	20	-7	64	68	58	62	279
14	20	-7	64	68	58	62	279
15	20	-7	64	68	58	62	279
16	20	-7	64	68	58	62	279
17	20	-7	64	68	58	62	279
18	20	-7	64	68	58	62	279
19	20	-7	64	68	58	62	279
20	19	-7	64	68	58	62	278
21	19	-7	64	68	58	62	278
22	19	-7	64	68	58	62	278
23	42	0	-10	67	57	63	239
24	42	0	-10	67	57	63	239
25	42	0	-10	67	57	63	239
26	42	0	-10	67	57	63	239
27	42	0	-10	67	57	63	239
28	42	0	-10	67	57	63	239
29	42	0	-10	67	57	63	239
30	42	0	-10	67	57	63	239

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
31	42	0	-10	67	57	63	239
32	42	0	-10	67	57	63	239
33	34	-9	3	66	58	63	233
34	34	-9	3	66	58	63	233
35	34	-9	3	66	58	63	233
36	34	-9	3	66	58	63	233
37	34	-9	3	66	58	63	233
38	34	-9	3	66	58	63	233
39	34	-9	3	66	58	63	233
40	18	-7	55	67	58	62	267
41	18	-7	55	67	58	62	267
42	18	-7	55	67	58	62	267
43	13	21	9	66	58	62	229
44	13	21	9	66	58	62	229
45	13	21	9	66	58	62	229
46	13	21	9	66	58	62	229
47	13	21	6	66	58	62	226
48	13	16	0	66	58	62	215
49	13	12	-2	66	58	62	213
50	13	8	-3	66	58	62	210
51	13	5	-3	66	58	62	207
52	13	2	-3	66	58	62	204
53	13	0	-3	66	58	62	202
54	13	0	-3	66	58	62	202
55	13	0	-3	66	58	62	202
56	13	0	-3	66	58	62	202
57	13	0	-3	66	58	62	202
58	12	0	-3	66	58	62	201
59	12	0	-2	66	58	62	200
60	12	0	-1	66	58	62	199
61	12	0	0	66	58	62	198
62	12	0	0	66	58	62	198

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
63	12	0	0	66	58	62	198
64	12	0	0	66	58	62	198
65	12	0	0	66	58	62	198
66	12	0	0	66	58	62	198
67	12	0	0	66	58	62	198
68	12	0	0	66	58	62	198
69	11	0	0	66	58	62	197
70	10	0	0	66	58	62	196
71	9	0	0	66	58	62	195
72	8	0	0	66	58	62	194
73	7	0	0	66	58	62	193
74	6	0	0	66	58	62	192
75	5	0	0	66	58	62	191
76	4	0	0	66	58	62	190
77	3	0	0	66	58	62	189
78	2	0	0	66	58	62	188
79	1	0	0	66	58	62	187
80	0	0	0	66	58	62	186
81	0	0	0	66	58	62	186
82	0	0	0	66	58	62	186
83	0	0	0	66	58	62	186
84	0	0	0	66	58	62	186
85	0	0	0	66	58	62	186
86	0	0	0	66	58	62	186
87	0	0	0	66	58	62	186
88	0	0	0	66	58	62	186
89	0	0	0	66	58	62	186
90	0	0	0	66	58	62	186
91	0	0	0	66	58	62	186
92	0	0	0	66	58	62	186
93	0	0	0	66	58	62	186
94	0	0	0	66	58	62	186

Gerações	Capacidade [MVar] instalada por barramento						Capacidade total [MVar]
	1	2	3	4	5	6	
95	0	0	0	66	58	62	186
96	0	0	0	66	58	62	186
97	0	0	0	66	58	62	186
98	0	0	0	66	58	62	186
99	0	0	0	66	58	62	186
100	0	0	0	66	58	62	186

Anexo II

Rede de seis barramentos, “IEEE_6BUS”.

Dados dos barramentos:

Barramento		Geração		Carga		Tensão
Número	Tipo	P [MW]	Q [MVar]	P [MW]	Q [MVar]	V [pu]
1	Balanço	108.034	19.783	0	0	1.04
2	Geração	50	94.160	0	0	1.04
3	Geração	60	68.088	0	0	1.04
4	Não geração	0	0	70	70	0.977
5	Não geração	0	0	70	70	0.966
6	Não geração	0	0	70	70	0.981

Dados das linhas:

De	Para	Resistência	Reatância	Suscetância	Rate A
		R [pu]	X [pu]	B [pu]	[MVA]
1	2	0.100	0.200	0.040	30
1	4	0.050	0.200	0.040	50
1	5	0.080	0.300	0.060	40
2	3	0.050	0.250	0.060	20
2	4	0.050	0.100	0.020	40
2	5	0.100	0.300	0.040	20
2	6	0.070	0.200	0.050	30
3	5	0.120	0.260	0.050	20
3	6	0.020	0.100	0.020	60
4	5	0.200	0.400	0.080	20
5	6	0.100	0.300	0.060	20

Anexo III

Rede de catorze barramentos, “IEEE14”.

Dados dos barramentos:

Barramento		Geração		Carga		Tensão
Número	Tipo	P [MW]	Q [MVar]	P [MW]	Q [MVar]	V [pu]
1	Balanço	232.966	-22.76	0	0	1.045
2	Geração	40	50	21.7	12.7	1.034
3	Geração	0	40	94.2	19	1.010
4	Não geração	0	0	47.8	-3.9	1.002
5	Não geração	0	0	7.6	1.6	1.004
6	Geração	0	10.681	11.2	7.5	1.045
7	Não geração	0	0	0	0	1.031
8	Geração	0	7.968	0	0	1.045
9	Não geração	0	0	29.5	16.6	1.028
10	Não geração	0	0	9	5.8	1.023
11	Não geração	0	0	3.5	1.8	1.031
12	Não geração	0	0	6.1	1.6	1.030
13	Não geração	0	0	13.5	5.8	1.025
14	Não geração	0	0	14.9	5	1.008

Dados das linhas:

De	Para	Resistência	Reatância	Suscetância	Rate A
		R [pu]	X [pu]	B [pu]	[MVA]
1	2	0.019	0.059	0.053	120
1	5	0.054	0.223	0.049	65
2	3	0.047	0.198	0.044	36
2	4	0.058	0.176	0.034	65
2	5	0.057	0.174	0.035	50
3	4	0.067	0.171	0.013	65
4	5	0.013	0.042	0.000	45
6	11	0.095	0.199	0.000	18
6	12	0.123	0.256	0.000	32
6	13	0.066	0.130	0.000	32
7	8	0.000	0.176	0.000	32
7	9	0.000	0.110	0.000	32
9	10	0.032	0.085	0.000	32
9	14	0.127	0.270	0.000	32
10	11	0.082	0.192	0.000	12
12	13	0.221	0.200	0.000	12
13	14	0.171	0.348	0.000	12

Anexo IV

Rede de vinte e um barramentos, “Rede_MAT_AT”.

Dados dos barramentos:

Barramento		Geração		Carga		Tensão
Número	Tipo	P [MW]	Q [MVar]	P [MW]	Q [MVar]	V [pu]
1	Balanço	302.154	132.167	0	0	1.040
2	Geração	150	82.249	0	0	1.040
3	Geração	150	82.248	0	0	1.040
4	Não geração	0	0	0	0	1.006
5	Não geração	0	0	0	0	1.007
6	Não geração	0	0	0	0	1.004
7	Não geração	0	0	0	0	1.005
8	Não geração	0	0	85	25	0.972
9	Não geração	0	0	95	20	0.974
10	Não geração	0	0	40	5	0.941
11	Não geração	0	0	40	10	0.932
12	Não geração	0	0	40	10	0.935
13	Não geração	0	0	40	15	0.914
14	Não geração	0	0	40	15	0.951
15	Não geração	0	0	3	0.8	0.955
16	Não geração	0	0	5	1	0.920
17	Não geração	0	0	47	13.5	0.954
18	Não geração	0	0	49	10.5	0.920
19	Não geração	0	0	49	15.5	0.921
20	Não geração	0	0	5	1	0.960
21	Não geração	0	0	52.5	11	0.945

Dados das linhas:

De	Para	Resistência	Reatância	Suscetância	Rate A
		R [pu]	X [pu]	B [pu]	[MVA]
4	5	0.000263	0.001799	0.000631	250
4	6	0.000263	0.001800	0.000631	250
4	7	0.000263	0.001800	0.000631	250
5	7	0.000263	0.001800	0.000631	250
6	7	0.000263	0.001800	0.000631	250
8	10	0.014310	0.041390	0.002210	160
8	12	0.039750	0.114960	0.006150	160
8	14	0.023850	0.068980	0.003690	160
9	15	0.012670	0.038000	0.000500	120
9	17	0.015830	0.047500	0.000630	80
9	19	0.044330	0.133000	0.001750	80
9	20	0.069670	0.209000	0.002750	80
10	11	0.015830	0.047500	0.000630	80
10	13	0.028500	0.085500	0.001130	80
11	12	0.015830	0.047500	0.000630	80
13	18	0.015830	0.047500	0.000630	50
14	16	0.015900	0.045980	0.002460	160
15	18	0.038000	0.114000	0.001500	80
15	21	0.022170	0.047500	0.000630	80
16	18	0.022170	0.066500	0.000880	80
16	19	0.022170	0.066500	0.000880	80
17	20	0.050670	0.152000	0.002000	80
17	21	0.022170	0.066500	0.000880	80

Anexo V

Tabela 1: Perdas totais da rede original de seis barramentos e da rede com implementação de bancos de condensadores.

De	Para	Sem implementação de bancos de condensadores		Com implementação de bancos de condensadores	
				<i>PSO e EPSO</i>	
		<i>P</i> [MW]	<i>Q</i> [MVar]	<i>P</i> [MW]	<i>Q</i> [MVar]
1	2	0.928	1.856	0.669	1.339
1	4	1.128	4.512	0.922	3.689
1	5	1.163	4.361	0.924	3.466
2	3	0.002	0.009	0.002	0.011
2	4	1.605	3.210	0.432	0.863
2	5	0.616	1.849	0.247	0.741
2	6	0.768	2.195	0.397	1.134
3	5	0.868	1.880	0.292	0.633
3	6	0.883	4.416	0.394	1.968
4	5	0.042	0.084	0.029	0.058
5	6	0.030	0.090	0.003	0.008
Perdas totais		8.033	24.462	4.311	13.911

Tabela 2: Perdas totais da rede original de catorze barramentos e das redes com implementação de bancos de condensadores.

De	Para	Sem implementação de bancos de condensadores		Com implementação de bancos de condensadores			
				<i>PSO</i>		<i>EPSO</i>	
		<i>P</i> [MW]	<i>Q</i> [MVar]	<i>P</i> [MW]	<i>Q</i> [MVar]	<i>P</i> [MW]	<i>Q</i> [MVar]
1	2	4.637	14.157	4.586	14.002	4.588	14.009
1	5	2.488	10.269	2.492	10.289	2.487	10.269
2	3	0.952	4.013	0.953	4.016	0.953	4.016
2	3	0.952	4.013	0.953	4.016	0.953	4.016
2	4	1.195	3.626	1.197	3.631	1.194	3.622
2	5	0.661	2.017	0.644	1.966	0.644	1.966
3	4	0.032	0.081	0.016	0.042	0.020	0.051
4	5	0.360	1.135	0.377	1.190	0.380	1.200
4	7	0.000	1.612	0.000	1.840	0.000	1.746
4	9	0.000	1.402	0.000	1.423	0.000	1.419
5	6	0.000	4.672	0.000	4.402	0.000	4.537
6	11	0.059	0.123	0.037	0.078	0.038	0.079
6	12	0.076	0.157	0.063	0.131	0.063	0.132
6	13	0.223	0.439	0.182	0.359	0.184	0.362
7	8	0.000	0.104	0.000	0.008	0.000	0.001
7	9	0.000	0.843	0.000	0.850	0.000	0.857
9	10	0.013	0.034	0.011	0.028	0.011	0.029
9	14	0.123	0.262	0.116	0.248	0.117	0.249
10	11	0.014	0.032	0.007	0.017	0.007	0.017
12	13	0.007	0.006	0.004	0.004	0.004	0.004
13	14	0.057	0.116	0.041	0.084	0.042	0.085
Perdas totais		11.847	49.114	11.681	48.623	11.687	48.666

Tabela 3: Perdas totais da rede original de vinte e um barramentos e das redes com implementação de bancos de condensadores.

De	Para	Sem implementação de bancos de condensadores		Com implementação de bancos de condensadores			
				<i>PSO</i>		<i>EPSO</i>	
		<i>P</i> [MW]	<i>Q</i> [MVA _r]	<i>P</i> [MW]	<i>Q</i> [MVA _r]	<i>P</i> [MW]	<i>Q</i> [MVA _r]
1	4	0.000	29.576	0.000	24.580	0.000	24.577
2	5	0.000	11.905	0.000	9.272	0.000	9.281
3	5	0.000	11.905	0.000	9.272	0.000	9.281
4	5	0.019	0.127	0.014	0.097	0.013	0.089
4	6	0.148	1.015	0.116	0.797	0.117	0.804
4	7	0.069	0.475	0.056	0.382	0.056	0.382
5	7	0.158	1.085	0.123	0.845	0.122	0.838
6	7	0.015	0.102	0.011	0.076	0.012	0.081
6	8	0.000	14.367	0.000	11.220	0.000	11.232
6	8	0.000	14.367	0.000	11.220	0.000	11.232
7	9	0.000	16.282	0.000	13.193	0.000	13.193
7	9	0.000	16.282	0.000	13.193	0.000	13.193
8	10	2.158	6.242	1.684	4.870	1.686	4.877
8	12	1.164	3.366	0.911	2.634	0.912	2.638
8	14	0.480	1.387	0.356	1.029	0.356	1.031
9	15	0.976	2.927	0.782	2.344	0.780	2.341
9	17	0.774	2.324	0.629	1.887	0.623	1.869
9	19	1.839	5.517	1.425	4.275	1.426	4.278
9	20	0.091	0.272	0.076	0.227	0.074	0.222
10	11	0.184	0.553	0.145	0.434	0.145	0.435

De	Para	Sem implementação de bancos de condensadores		Com implementação de bancos de condensadores			
				<i>PSO</i>		<i>EPSO</i>	
		<i>P</i> [MW]	<i>Q</i> [MVar]	<i>P</i> [MW]	<i>Q</i> [MVar]	<i>P</i> [MW]	<i>Q</i> [MVar]
10	13	0.622	1.865	0.441	1.323	0.442	1.325
11	12	0.016	0.048	0.012	0.037	0.012	0.037
13	18	0.000	0.000	0.000	0.000	0.000	0.000
14	16	0.000	0.000	0.000	0.000	0.000	0.000
15	18	1.038	3.115	0.810	2.431	0.812	2.436
15	21	0.233	0.499	0.192	0.412	0.192	0.411
16	18	0.002	0.007	0.002	0.005	0.002	0.006
16	19	0.016	0.048	0.013	0.040	0.013	0.040
17	20	0.019	0.058	0.015	0.045	0.015	0.045
17	21	0.132	0.396	0.106	0.319	0.108	0.322
Perdas totais		10.154	146.111	7.920	116.462	7.919	116.499

Anexo VI

Módulo do algoritmo PSO.

Ficheiro `pso_swarm`: onde estão definidas as funções `create_swarm` (responsável por criar enxame) e `update_swarm` (atualiza o enxame com o auxílio de outras funções).

```
import redirect, excelpy, psspy

import pso_particle
import constants

def create_swarm():
    s=[0]*constants.num_p

    for i in range (constants.num_p):
        part=pso_particle.create_particle()
        s[i]=part

    return s

def update_swarm( s, best):

    best_of= (len(best))*[100]

    for i in range(len(s)):
        #evaluate
        s[i], best, best_of=pso_particle.evaluate_particle(s[i], best_of,
best)
    for i in range(len(s)):
        #movement
        s[i]=pso_particle.movement(s[i], best)
    return s, best, best_of
```

Ficheiro pso_particle: onde estão definidas as funções *create_particle* (responsável por criar partícula); *movement* (responsável pelo movimento da partícula); *evaluate_particle* (avalia a aptidão da partícula).

```
import redirect, excelpy, psspy

import psse_function
import constants
import random

def create_particle():
    part = (constants.size_p*3+1)*[0]

    for i in range(constants.size_p):
        for j in range(len(constants.bus_loads)/2):
            if i+1==constants.bus_loads[j] :
                cap=abs(int(constants.bus_loads[j+(len(constants.bus_loads)/2)]))
                part[i] = random.randint(-cap, cap)
            elif j+1==len(constants.bus_loads)/2:
                part[i] = random.randint(constants.min_cap, constants.max_cap)

        part[i+constants.size_p] = part[i]
        part[i+constants.size_p*2]= random.randint(constants.v_min, constants.v_max)
        psse_function.create_shunt(i+1, '2', part[i])

    psse_function.solve_case()
    part[constants.size_p*3]= psse_function.get_from_branch("PLOSS")

    return part

def movement (part, best):

    dim= (len(part)-1)/3

    for i in range(dim):
        #  $V(k+1) = w*V(k) + c1*rand*(X_{best}-X(k)) + c2*rand*(X_{global}-X(k))$ 
        part[i+dim*2] = constants.w*part[i+dim*2] +
        constants.c1*random.random()*(part[i+dim]-part[i]) +
        constants.c2*random.random()*(best[i]-part[i])

        if part[i+dim*2] > constants.v_max:
            part[i+dim*2]= constants.v_max
        if part[i+dim*2] < constants.v_min:
            part[i+dim*2]= constants.v_min

    position = part[i] + part[i+dim*2]

    for j in range(len(constants.bus_loads)/2):
        cap=abs(int(constants.bus_loads[j+(len(constants.bus_loads)/2)]))
        if i+1==constants.bus_loads[j]:
            if position > cap:
                position=cap
            part[i+dim*2]= -part[i+dim*2]
```

```

        elif position < -cap:
            position=-cap
            part[i+dim*2]= -part[i+dim*2]
    elif j+1==len(constants.bus_loads)/2:
        if position > constants.max_cap:
            position = constants.max_cap
            part[i+dim*2]= -part[i+dim*2] #bounce
        if position < constants.min_cap:
            position = constants.min_cap
            part[i+dim*2]= -part[i+dim*2] #bounce

    part[i] = int(position)

    if part[i+dim*2] > constants.v_max:
        part[i+dim*2]= constants.v_max
    if part[i+dim*2] < constants.v_min:
        part[i+dim*2]= constants.v_min

    return part

def evaluate_particle(part, best_of, best):

    capacity=0
    capacity_particle = 0
    capacity_best_of = 0
    capacity_best = 0
    dim= (len(part)-1)/3
    fitness=0

    for i in range(dim):
        psse_function.change_shunt(i+1, '2', 1, part[i])
        capacity += abs(part[i])
        capacity_particle += abs(part[i+dim])
        capacity_best_of += abs(best_of[i])
        capacity_best += abs(best[i])

    psse_function.solve_case()
    fitness += psse_function.get_from_branch("PLOSS")

    #check if particle is acceptable (busvoltage)
    voltage= psse_function.bus_voltage()
    for k in range(len(voltage)):
        if voltage[k] < constants.max_voltage and voltage[k] >
constants.min_voltage:
            if k == (len(voltage)-1):

                #check if it's the best of particle
                if abs(fitness - part[dim*3]) < constants.dif_capacity and capacity <
capacity_particle or fitness < part[dim*3]: #Ploss diference less then 0.1kW and
less MVar capacity
                    part[dim*3] = fitness
                    for i in range(dim):
                        part[i+dim] = part[i]

                #check if it's the best of iteration
                if abs(fitness - best_of[dim]) < constants.dif_capacity and capacity <
capacity_best_of or fitness < best_of[dim]:
                    best_of[dim] = fitness

```

```

        for i in range(dim):
            best_of[i] = part[i]

        #check if it's the best of all
        if abs(fitness - best[dim]) < constants.dif_capacity and capacity <
capacity_best or fitness < best[dim]:
            best[dim] = fitness
            for i in range(dim):
                best[i] = part[i]

    else:
        #if voltage isn't between interval then break
        break

return part, best, best_of

```

Módulo do algoritmo PSO.

Ficheiro `epso_swarm`: onde estão definidas as funções `create_swarm` (responsável por criar enxame) e `update_swarm` (atualiza o enxame com o auxílio de outras funções).

```
import sys, os, redirect, excelpy, psspy

import epso_particle
import random
import constants

def create_swarm():
    # Enxame - num_p particulas de size_p dimensoes
    s=[0]*constants.num_p

    for i in range (constants.num_p):
        s[i]=epso_particle.create_particle()

    W = [[3*[random.uniform(0,2)] for i in range (constants.size_p)] for i in range(constants.num_p)]
    Wbest= [random.random()*constants.size_p

    return s, W, Wbest,

def update_swarm( s, W, Wbest, best):

    best_of= len(best)*[1000]
    new_best= (constants.size_p)*[0]
    Wbest_of = [3*[0] for i in range (len(W[0]))]

    for i in range(len(s)):
        #mutation best
        Wbest=epso_particle.mutation(Wbest, constants.tau)

        for b in range(len(new_best)):
            new_best[b]=best[b]
            new_best=epso_particle.new_bg(new_best, Wbest)

    #####replication
    repl,Wrepl=epso_particle.replication(s[i], W[i], constants.r)

    #####mutation
    for j in range(constants.r):
        for k in range(len(Wrepl[0])):
            Wrepl[j][k]=epso_particle.mutation(Wrepl[j][k], constants.tau)

    #####movement and fitness
    #movement and fitness original
    part=epso_particle.movement(s[i], W[i], new_best)
    part[constants.size_p*3+1]=epso_particle.fitness(part)
    #movement and fitness replicas
    for j in range(constants.r):
        108
```

```

        repl[j]=epso_particle.movement(repl[j], Wrepl[j], new_best)
        repl[j][constants.size_p*3+1]=epso_particle.fitness(repl[j])

#####selection
        s[i], W[i]=epso_particle.selection_particle(part, repl, W[i], Wrepl)

        for i in range(len(s)):
            part, best, best_of, Wbest_of=epso_particle.update_best(s[i], W[i], best,
            best_of, Wbest_of)

        return s, best, best_of, Wbest_of

```

Ficheiro `epso_particle`: onde estão definidas as funções *create_particle* (responsável por criar partícula); *replication* (responsável por replicar a partícula); *mutation* (responsável pela mutação dos parâmetros estratégicos); *new_bg* (responsável pela mutação da melhor solução encontrada até ao momento); *movement* (responsável pelo movimento da partícula); *fitness* (avalia a aptidão da partícula); *selection_particle* (seleciona a melhor solução do conjunto constituído pela partícula original e suas réplicas); *update_best* (atualiza a melhor solução encontrada até ao momento e da geração).

```

import sys, os, redirect, psspy
redirect.psse2py()

import psse_function
import random

import constants

def create_particle():
    # Enxame - num_p particulas de size_p dimensoes + 2 (fit da actual e fit da
    melhor encontrada)
    part = (constants.size_p*3+2)*[0]

    for i in range(constants.size_p):
        for j in range(len(constants.bus_loads)/2):
            if i+1==constants.bus_loads[j] :
                cap=abs(int(constants.bus_loads[j+(len(constants.bus_loads)/2)]))
                part[i] = random.randint(-cap, cap)
            elif j+1==len(constants.bus_loads)/2:
                part[i] = random.randint(constants.min_cap, constants.max_cap)

        part[i+constants.size_p] = part[i]
        part[i+constants.size_p*2]= random.randint(constants.v_min, constants.v_max)
        psse_function.create_shunt(i+1, '2', part[i])

    part[constants.size_p*3+1]=fitness(part)
    part[constants.size_p*3]=part[constants.size_p*3+1]

    return part

```

```

def replication(part,W,r):
    # do r replicas of particle repl=[[part],[part], ... ]
    repl= [[part[i] for i in range(len(part))] for j in range(r)]
    Wrepl=[[W[i] for i in range(len(W))] for k in range(r)]
    return repl, Wrepl

def mutation (w,tau):

    for i in range (len(w)):
        w[i]=w[i] * (1 + tau * random.gauss(0,1))
        if w[i]<0:
            w[i]=0
        if w[i]>2:
            w[i]=2

    return w

def new_bg(bestaux, Wbest):

    for i in range(len(Wbest)):
        bestaux[i]= bestaux[i] + Wbest[i]*random.gauss(0,1)
        if bestaux[i]> constants.max_cap:
            bestaux[i]= constants.max_cap
        if bestaux[i]< constants.min_cap:
            bestaux[i]= constants.min_cap

    return bestaux

def movement (part, W, best):

    dim= (len(part)-2)/3

    for i in range(dim):
        #  $V(k+1) = W_{inercia} * V(k) + W_{memoria} * (b_i - X(k)) + W_{cooperacao} * (bG(novo) - X(k))$ 
        #if random number is under p then past global best enter in movement
        if random.random()<=constants.p:
            part[i+dim*2] = W[i][0]*part[i+dim*2] + W[i][1]*(part[i+dim]-part[i]) +
W[i][2]*(best[i]-part[i])
        else:
            part[i+dim*2] = W[i][0]*part[i+dim*2] + W[i][1]*(part[i+dim]-part[i])

        if part[i+dim*2] > constants.v_max:
            part[i+dim*2]= constants.v_max
        if part[i+dim*2] < constants.v_min:
            part[i+dim*2]= constants.v_min

        position = part[i] + part[i+dim*2]

        for j in range(len(constants.bus_loads)/2):
            cap=abs(int(constants.bus_loads[j+(len(constants.bus_loads)/2)]))
            if i+1==constants.bus_loads[j]:
                if position >cap:
                    position=cap
                    part[i+dim*2]= -part[i+dim*2]
                elif position < -cap:

```



```

        position=-cap
        part[i+dim*2]= -part[i+dim*2]
    elif j+1==len(constants.bus_loads)/2:
        if position > constants.max_cap:
            position = constants.max_cap
            part[i+dim*2]= -part[i+dim*2] #bounce
        if position < constants.min_cap:
            position = constants.min_cap
            part[i+dim*2]= -part[i+dim*2] #bounce

    part[i] = int(position)

    #if vmax=-vmin
    if part[i+dim*2] > constants.v_max:
        part[i+dim*2]= constants.v_max
    if part[i+dim*2] < constants.v_min:
        part[i+dim*2]= constants.v_min

    return part

def fitness (part):

    dim=(len(part)-2)/3
    fitness=0

    for i in range(dim):
        # if there is overcompensation then particle receive a penalty
        for j in range(len(constants.bus_loads)/2):
            if i+1==constants.bus_loads[j] and
abs(part[i])>abs(constants.bus_loads[j+(len(constants.bus_loads)/2)]):
                fitness += constants.penalty_compensation

        psse_function.change_shunt(i+1, '2', 1, part[i])

    psse_function.solve_case()
    #check if particle is acceptable (busvoltage) if not fitness=False
    voltage= psse_function.bus_voltage()
    for k in range(len(voltage)):
        if voltage[k] < constants.max_voltage and voltage [k] >
constants.min_voltage:
            if k == (len(voltage)-1):
                fitness += psse_function.get_from_branch("PLOSS")
            else:
                fitness =constants.penalty_voltage
                break
    return fitness

def selection_particle(part, repl, W, Wrepl):
    # selection elitist

    repl_capacity = (len(repl))*[0]
    dim= (len(part)-2)/3
    bestaux1=part
    Wbestaux=W
    best_capacity=0
    for r in range(len(repl)):
        for i in range(dim):
            if r ==0:
                best_capacity += abs(part[i])
                repl_capacity[r] += abs(repl[r][i])

```

```

        else:
            repl_capacity[r] += abs(repl[r][i])

    for r in range(len(repl)):
        if repl[r][dim*3+1] < bestaux1[dim*3+1] and abs(bestaux1[dim*3+1] -
            repl[r][dim*3+1]) >= constants.dif_capacity or abs(bestaux1[dim*3+1] -
            repl[r][dim*3+1]) < constants.dif_capacity and repl_capacity[r] < best_capacity:
            bestaux1=repl[r]
            best_capacity= repl_capacity[r]
            Wbestaux=Wrepl[r]

    return bestaux1, Wbestaux

def update_best (part, W, best, best_of, Wbest_of):

    dim= (len(part)-2)/3
    cap=0
    capacity_particle = 0
    capacity_best_of = 0
    capacity_best = 0

    for i in range(dim):
        cap += abs(part[i])
        capacity_particle += abs(part[dim+i])
        capacity_best_of += abs(best_of[i])
        capacity_best += abs(best[i])

    #check if it's the best of particle
    if part[dim*3+1] < part[dim*3] and abs(part[dim*3+1] - part[dim*3]) >=
    constants.dif_capacity or abs(part[dim*3+1] - part[dim*3]) <
    constants.dif_capacity and cap < capacity_particle:
        part[dim*3] = part[dim*3+1]
        for i in range(dim):
            part[i+dim] = part[i]

    #check if it's the best of iteration
    if part[dim*3+1] < best_of[dim] and abs(part[dim*3+1] - best_of[dim]) >=
    constants.dif_capacity or abs(part[dim*3+1] - best_of[dim]) <
    constants.dif_capacity and cap < capacity_best_of:
        best_of[dim] = part[dim*3+1]
        Wbest_of=W
        for i in range(dim):
            best_of[i] = part[i]

    #check if it's the best of all
    if part[dim*3+1] < best[dim] and abs(part[dim*3+1] - best[dim]) >=
    constants.dif_capacity or abs(part[dim*3+1] - best[dim]) < constants.dif_capacity
    and cap < capacity_best:
        best[dim] = part[dim*3+1]
        for i in range(dim):
            best[i] = part[i]

    return part, best, best_of, Wbest_of

```